QLogic 12000 Series InfiniBand Switches and Directors

CLI Reference Guide

Version 6.0

Document Revision History	
Revision E, August 15, 2010	
Changes	**Sections Affected**
Added the following commands to the group Network: dnsParamsShow dnsParamsSet	"Network" on page 1-39

Table of Contents

Preface

This manual describes the command line interface (CLI) task information for the QLogic 12300 and 12800 family of switches.

This manual is organized as follows:

This section describes the intended audience and technical support.

Section 1 describes the 12000 switch CLI commands.

Intended Audience

This manual is intended to provide network administrators and other qualified personnel a reference for the command line interface of the QLogic 12000 series switches.

Documentation Conventions

This guide uses the following documentation conventions:

- **NOTE:** provides additional information.

- *CAUTION!* indicates the presence of a hazard that has the potential of causing damage to data or equipment.

- *WARNING!!* indicates the presence of a hazard that has the potential of causing personal injury.

- Text in blue font indicates a hyperlink (jump) to a figure, table, or section in this guide, and links to Web sites are shown in underlined blue. For example:

  ❑ Table 9-2 lists problems related to the user interface and remote agent.

  ❑ See "Installation Checklist" on page 3-6.

  ❑ For more information, visit www.qlogic.com.

- Text in **bold** font indicates user interface elements such as a menu items, buttons, check boxes, or column headings. For example:

  ❑ Click the **Start** button, point to **Programs**, point to **Accessories**, and then click **Command Prompt**.

  ❑ Under **Notification Options**, select the **Warning Alarms** check box.

■ Text in `Courier` font indicates a file name, directory path, or command line text. For example:

❑ To return to the root directory from anywhere in the file structure: Type `cd /root` and press ENTER.

❑ Enter the following command: `sh ./install.bin`

■ Key names and key strokes are indicated with UPPERCASE:

❑ Press CTRL+P.

❑ Press the UP ARROW key.

■ Text in *italics* indicates terms, emphasis, variables, or document titles. For example:

❑ For a complete listing of license agreements, refer to the *QLogic Software End User License Agreement*.

❑ What are *shortcut keys*?

❑ To enter the date type *mm/dd/yyyy* (where *mm* is the month, *dd* is the day, and *yyyy* is the year).

■ Topic titles between quotation marks identify related topics either within this manual or in the online help, which is also referred to as *the help system* throughout this document.

License Agreements

Refer to the *QLogic Software End User License Agreement* for a complete listing of all license agreements affecting this product.

Technical Support

Customers should contact their authorized maintenance provider for technical support of their QLogic products. QLogic-direct customers may contact QLogic Technical Support; others will be redirected to their authorized maintenance provider.

Visit the QLogic support Web site listed in Contact Information for the latest firmware and software updates.

Availability

QLogic Technical Support for products under warranty is available during local standard working hours excluding QLogic Observed Holidays.

Training

QLogic offers training for technical professionals for all iSCSI, InfiniBand, and Fibre Channel products. From the main QLogic web page at www.qlogic.com, click the **Education and Resources** tab at the top, then click the **Education & Training** tab on the left. The QLogic Global Training Portal offers online courses, certification exams, and scheduling of in-person training.

Technical Certification courses include installation, maintenance and troubleshooting QLogic products. Upon demonstrating knowledge using live equipment, QLogic awards a certificate identifying the student as a Certified Professional. The training professionals at QLogic may be reached by e-mail at training@qlogic.com.

Contact Information

Please feel free to contact your QLogic approved reseller or QLogic Technical Support at any phase of integration for assistance. QLogic Technical Support can be reached by the following methods:

Web	http://support.qlogic.com
Email	support@qlogic.com

Knowledge Database

The QLogic knowledge database is an extensive collection of QLogic product information that you can search for specific solutions. We are constantly adding to the collection of information in our database to provide answers to your most urgent questions. Access the database from the QLogic Support Center: http://support.qlogic.com.

![QLogic — The Ultimate in Performance]

1 Switch Command Line Interface

Overview

This section details the usage of the Command Line Interface (CLI) feature for the QLogic 12000 family of switches:

■   The QLogic 12300

■   The QLogic 12800 Series:

❑   12800-040

❑   12800-120

❑   12800-180

❑   12800-360

The CLI allows the user to perform remote configuration and management tasks, which in many respects mirrors the functionality of the Chassis Viewer GUI.

The CLI is accessed via a terminal attached to the RS232 port(s) or via the OOB management port using Telnet and secure shell (SSH) of:

■   The 12300 chassis

■   The SEEB module(s) of the 12800 series

Accessing the CLI through the serial port does not require a login and password (unless configured to do so) and defaults to administrator privileges. Providing access through the serial port means that users will always have access to the switch, even if Telnet and SSH are not functioning.

For a standalone switch, the user would Telnet to the IP address(es) of the unit. Once connected, the CLI works as any telnet session does.

To access the CLI using Telnet and SSH, a login and password is required. There are two user modes, operator and administrator with the following access privileges:

Operator:

■   Read only access.

Administrator:

- Read and write Access.

- Reboot access.

- Can change operator and administrator passwords.

- Can disable user login and passwords. This would allow all users admin-level access without the need for a user name or password.

- Can view all current user sessions

- Can access all of the commands executed from any open operator session.

- Can log out any open user sessions

- Can send messages to the open user sessions

The CLI allows multiple users to be logged in simultaneously. However, some commands will be locked to a user(s) if another user is executing the same command.

Commands and Functional Groups

The list of available commands can be accessed by typing **list**. To keep the list short, the commands are grouped into functional groups, which are:

General:

General commands for user management and CLI configuration.

Deprecated:

Commands that have been deprecated.

The Deprecated group contains CLI commands that have been replaced or are to be removed. Please use the new command where appropriate.

Chassis:

Chassis management commands. (FRU info, fan/power supply state, etc).

Network:

Ethernet interface management commands.

Firmware:

Provides commands for updating the firmware via a File Transfer Protocol (FTP) server or Secure Copy Protocol (SCP) (if using SSH to access the Bridge Module). The switch has the ability to store the location of the firmware files for future upgrades. Additionally, the Firmware functional group includes commands for viewing the current firmware revisions and for changing the boot image.

SubnetManagement:

InfiniBand subnet manager configuration and management.

Log:

Provides commands for viewing log files as well as configuring logging parameters.

KeyManagement:

License key management.

IbSwitchInfo:

Provides commands for displaying InfiniBand (IB) statistics for all IB ports on the switch, as well as for configuring port statistic thresholds.

TimeManagement:

Provides commands for retrieving and setting the current system time, as well as commands for setting the time zone and daylight saving time parameters.

Snmp:

Provides commands for configuring SNMP trap destinations and security parameters required to access the switch from an SNMP manager.

CaptureInfo:

Provides commands for capturing switch-specific information for the purposes of analysis and debugging.

To list commands within a functional group, simply type in the functional group name. For example, to list all of the firmware commands, type **list Firmware**. The system displays information similar to the following:

```
-> list Firmware
fwUpdateSlot        Update units firmware
fwUpdateChassis     Update units firmware
fwListFiles         List the contents of the firmware ramdisk
fwShowUpdateParams  Display firmware default update parameters
fwSetUpdateParams   Configure firmware default update parameters
showCapability      Display the capabilities/features
showLastScpRetCode  Display the return code from the last SCP Firmware Push
fwVersion           Display Firmware revisions
bootQuery           Query boot image information
bootSelect          Change boot selection
```

Online Help

The online help for the CLI provides, for each command, all necessary information to successfully execute the command. For example, typing **help list** (can also type **list help**) displays the following information for the `list` command:

```
NAME
   list
SYNOPSIS
   list [group] [-noprompt] [-verbose]
DESCRIPTION
   List available commands.
OPTIONS
   group     - List the commands in that particular group
   -noprompt - Just list the command groups.
   -verbose  - Print full help for each command, instead of summary.
NOTES
   Specify the group name or use 'all' to list all available commands.
```

Keyboard Shortcuts

■ The CLI keeps a history of recently executed commands. This history is available via the **Up** and **Down** arrow keys.

■ Users may edit the current command with the **Left** and **Right** arrow keys.

■ Tab completion: pressing the **Tab** key after typing at least one character either completes a command or lists all the available commands that begin with the characters already typed.

Accessing the CLI

NOTE: The CLI can be accessed via Telnet, SSH, or through the switch RS232 serial port. The following instructions use Telnet.

1. Telnet to the IP address of the switch (the default IP address is 192.168.100.9) with the following command:

   telnet <IP ADDRESS>

2. The system prompts for a username. The CLI has the following default user names:

    Operator access: **operator**

    Administrator access: **admin**

   Type the appropriate username and press **Enter**.

3. The system prompts for a password. The CLI has the following default passwords:

Operator access: **operpass**

Administrator access: **adminpass**

Type the appropriate password and press **Enter**. The system responds with:

```
Welcome to the <SWITCH> CLI. Type 'list' for the list of
commands.
```

Groups and Commands

The following section lists all CLI functional groups along with the commands for each group. Commands for all 12000 switches are listed. Any commands specific to a switch(es) is noted. For more specific information for each functional group, the user would execute the **help <GROUP NAME>** command. For more specific command information, the user would execute the **help <COMMAND NAME>** command.

General

help

Displays help information for a specific command.

Syntax:

```
help [command]
```

Options:

```
[command]
   The command to display help for.
```

Sample Output:

```
-> help list

NAME
      list

SYNOPSIS
      list [group] [-noprompt]

DESCRIPTION
      List available commands.

OPTIONS
      group     - List the commands in that particular group

      -noprompt - Just list the command groups.
```

NOTES: General Help

Type list or ? for the list of commands.

To get help on a particular command type: commandname help.

For convienence purposes you can also type: help commandname

Use the Up and Down arrow keys to browse command history, Left and Right arrow keys to edit the current command and the Tab key for tab completion of a command.

Two alternate key bindings exist for the backspace and delete keys. If these keys are not responding as expected use the **swapBsDel** command to swap the bindings.

Commands are grouped into subcategories. To list the commands in a subcategory type in the category heading. Category headings are identified by starting with a capital letter. For example, to list all the commands that handle log configuration type **Log**.

In each help description, items in brackets(e.g. [<command>] in this help description) represent optional parameters. User selectable input is indicated by items within the '<' and '>' symbols(e.g. <command>). Text outside the '<' and '>' characters is the actual text that needs to be entered. When there is more than one choice, the options are specified within '{' and '}' characters, where the options are separated by '|' characters.

list

Displays a list of all valid commands.

Syntax:

```
list [all] [group] [-noprompt] [-verbose]
```

Options:

```
all
```
List the commands for all groups.
```
group
```
Displays a list of commands for a particular group
```
-noprompt
```
Displays a list of the command groups only.
```
-verbose
```
Print full help for each command, instead of summary.

Sample Output:

```
list
List of Valid Commands:
General        General commands for user management and CLI configuration.
Deprecated     These commands have been deprecated
Network        Snmp configuration commands.
Firmware       Update firmware and display current revision levels.
Log            Log file display and configuration
IbSwitchInfo   InfiniBand port configuration and statistics
TimeManagement Display and configure the system time
Snmp           Snmp configuration commands.

Type the name of the group you want to list or return to exit:
```

NOTES:
Use 'list all' to display brief help for all available commands.

Use 'list all -verbose' to display verbose help for all commands.

Use 'list -noprompt' to display the list of command groups.

history

Displays the command history for this session.

Syntax:

```
history
```

Options:

```
None.
```

Sample Output:

```
-> history
command history [30 max lines]:
  list
  Deprecated
  Network
  Firmware
  Log
  IbSwitchInfo
  TimeManagement
  Snmp
  Ethernet
  help
  list
  history
```

reboot (12300)

Reboots the device.

Syntax:

```
reboot [now][-m][slot n][-s][-n][all]
```

Options:

```
now
    Does not prompt the user before rebooting.
-m
    Reboot Master (local) QLogic 12300 (non-disruptive).
slot n
    reset specific device where n = slotNumber (disruptive).
-s
    Reboot Slave (remote) QLogic 12300
-n
    Reboot Slave (remote)  management card only (non-disruptive)
all
    Reboot All local devices (excludes -n/-m/-s).
```

Sample Output:

```
-> reboot
Disruptive reboot selected
Proceed with reboot? [N]
```

> **NOTES:** You may reboot the local (Master) Managment Module or the remote (Slave) Managment Module with one or multiple arguments.
>
> Non-disruptive arguments will not interfere with switch traffic.
>
> Providing the argument **all** performs disruptive reboot of all present Managment Modules and cards.
>
> If rebooting the local device from telnet, ssh, etc., you will have to reconnect after rebooting.
>
> Default (no arguments) reboots local device disruptively after prompt.
>
> See also: 'resetCard', 'showInventory'.

reboot (12800 Series)

Reboots the device.

Syntax:

```
reboot [now][-m][slot n][-s][-n][all]
```

Options:

```
now
```
   Does not prompt the user before rebooting.
```
-m
```
   Reboot Master (local) Management Module 228 (non-disruptive).
```
slot n
```
   reset specific device where n = slotNumber (disruptive).
```
-s
```
   Reboot Slave (remote) Management Module 227.
```
-n
```
   Reboot Slave (remote) managment card only (non-disruptive)
```
all
```
   Reboot All local devices (excludes -n/-m/-s).

Sample Output:

```
-> reboot
Disruptive reboot selected
Proceed with reboot? [N]
```

> **NOTES:** You may reboot the local (Master) Managment Module or the remote (Slave) Managment Module with one or multiple arguments.
>
> Non-disruptive arguments will not interfere with switch traffic.
>
> Providing the argument **all** performs disruptive reboot of all present Managment Modules and cards.
>
> If rebooting the local device from telnet, ssh, etc., you will have to reconnect after rebooting.
>
> Default (no arguments) reboots local device disruptively after prompt.
>
> See also: 'resetCard', 'showInventory'.

killCliSession

   Terminates an existing CLI session

   Syntax:

```
killCliSession sessionNumber
```

   Options:

```
sessionNumber
```
   The sessionNumber that is returned from the **who** command.

Sample Output:

```
-> killCliSession
must supply session number
```

NOTES: This command logs out remote sessions. Use 'who' to obtain the list of active sessions.

who

Displays currently active CLI sessions.

Syntax:

```
who
```

Options:

```
None.
```

Sample Output:

```
user            role      index   logged in          last cmd            type    ip address
-----------------------------------------------------------------------------------------------
admin           admin     0       13:19:03 07/11/2008 09:41:20 07/11/2008 serial  0.0.0.0
```

NOTES: Displays a list of currently active CLI sessions. Note that a session can be 'active', but no user information available (in most cases, this indicates the session is waiting for the user to enter login information).

For each session the following information is displayed:

- user: username of the logged in user

- role: security role of the user

- index: internal session index

- logged in: timestamp of when the user logged in

- last cmd: timestamp of the users last command

- type: method used to connect to the system

- ip address: ip address of the user (if applicable)

broadcast

Write a message to all active CLI sessions

Syntax:

```
broadcast msg
```

Options:

```
msg
```
　　Message (Note: the message text must be encapsulated in quotes "".

Sample Output:

```
-> broadcast "The system will be rebooted in 5 minutes."
```

> **NOTES:** Writes the supplied message to all other active CLI sessions. The message to be written must be encapsulated in quotes, and non-empty.

swapBsDel

Swaps the backspace and delete character key bindings.

Syntax:

```
swapBsDel
```

Options:

　　None.

Sample Output:

```
-> swapBsDel
```

> **NOTES:** Terminals may bind the backspace and delete key bindings differently. This command swaps two commonly used bindings, which allow the user to use the backspace and delete keys properly without having to adjust their terminal settings. Backspace and delete swapping is persistently maintained per user (i.e. each login account can have a separate binding).

setTermWidth

Change the terminal width for text formatting purposes.

Syntax:

```
setTermWidth width
```

Options:

```
width
```
　　Width of the user's terminal window. Minimum width is 20 characters.

Sample Output:

```
-> setTermWidth 100
```

> **NOTES:** Allows modification of the terminal width used for text formatting purposes. Note that not all commands adhere to this setting. The minimum width is 20 characters.

getTermWidth

Displays the terminal width for text formatting purposes.

Syntax:

```
getTermWidth
```

Options:

```
None.
```

Sample Output:

```
-> getTermWidth
Current terminal width: 80 characters.
```

> **NOTES:** Displays the terminal width used for text formatting purposes. Note that not all commands adhere to this setting.

prompt

Set the CLI prompt (global for all active CLI sessions).

Syntax:

```
prompt str
```

Options:

```
str
```
  The new prompt.

Sample Output:

```
prompt -> "12300->"
12300->
```

> **NOTES:** This changes the prompt for all CLI sessions. The prompt may not exceed 11 characters and is not saved across reboots. If the prompt contains a space, asterick, comma, parenthesis or semicolon it must be enclosed with double quotes

". For example: "*a prompt*". Also, if a prompt is not accepted try to enclose it with double quotes. In order for some FastFabric Tools to function correctly, the prompt must end in "-> " (note trailing space).

case

Display or change the case sensitivity for the command interpreter for the CLI session.

Syntax:

```
case [off | on]
```

Options:

```
off | on
```
Turn case sensitivity off or on.

Sample Output:

```
-> case off
Case sensitivity is now off
```

> **NOTES:** This displays or changes the case sensitivity for the command interpreter for the CLI session. If an option is not specified the current case sensitivity is displayed, otherwise the case sensitivity is turned on or off depending on the specified option. When case sensitivity is on, the CLI input must match the exact character case (lower and upper case) as specified in the help text. When case sensitivity is turned off, the input may be any combination of upper and lower case.

showLastRetCode

Display the return code from the last executed command.

Syntax:

```
showLastRetCode [-brief]
```

Options:

```
-brief
```
Displays just the numeric value of the return code.

Sample Output:

```
-> showLastRetCode
Last Exit Code: 0: Success
```

> **NOTES:** This allows for automated systems to determine if a command was successful or not.

echo

Echoes text back to the output..

Syntax:

```
echo [<text> ...]
```

Options:

```
[<text> ...]
```
Text to be echoed..

Sample Output:

```
-> echo test
test
```

> **NOTES:** This echoes the text for each argument back to the output. If there are multiple arguments they are separated by spaces.

rlogin

Creates I/O terminal to local cards.

Syntax:

```
rlogin [hostName]
```

Options:

```
hostName
```
The name of the device to connect.

Sample Output:

```
-> rlogin Slave
```

> **NOTES:** This command allows users to open a terminal to local I/O devices within the chassis. The command rcmd is an alias of rlogin. See also hostShow.

rcmd

Creates I/O terminal to local cards.

Syntax:

```
rcmd [hostName]
```

Options:

```
hostName
```
The name of the device to connect.

Sample Output:

```
-> rcmd Slave
```

> **NOTES:** This command allows users to open a terminal to local I/O devices within the chassis. The command rcmd is an alias of rlogin. See also hostShow.

resetCard

Reboot a specific slot in the chassis.

Syntax:

```
resetCard n [now]
```

Options:

```
n
```
Chassis slot number.
```
now
```
No prompt before resetting the card.

Sample Output:

```
-> resetCard 101 now
```

> **NOTES:** If you reset the switch you are connected to you will have to reconnect. You can find out the valid slots for this device with the 'showInventory' command.

logout

Logout of the current CLI session.

Syntax:

```
logout
```

Options:

```
None.
```

Sample Output:

```
logout
```

user

Change user accounts.

Syntax:

```
user [user]
```

Options:

```
[user]
```
Name of user account to change to.

Sample Output:

```
-> user operator
User changed to: operator
```

> **NOTES:** Use this command to change to the 'operator' account, or to the 'admin' account. The minimum password length is 8 characters. The maximum length is 40 characters.

passwd

Change users password(s).

Syntax:

```
passwd [username]
```

Options:

```
[username]
```
Name of the user whose password needs to be changed.

Sample Output:

```
-> passwd operator1
User password changed successfully
```

NOTES: Allows the modification of the user's password.

userAdd

Add a user account.

Syntax:

```
userAdd [role] [username] [password]
```

Options:

```
[role]
```
Can be either **admin** or **operator**.

```
[username]
```
The name of the user to be added.

```
[password]
```
Optional password. If not supplied the default password for that role is used.

Sample Output:

```
-> userAdd admin Bob
User added: Bob
Password is set to the default password for this role: admin
```

userRem

Remove a user account.

Syntax:

```
userRem [username]
```

Options:

```
username
```
Name of the user to be removed.

Sample Output:

```
-> userRem Bob
User deleted: Bob
```

userListShow

List all user accounts for this device.

Syntax:

```
userListShow
```

Options:

None.

Sample Output:

```
-> userListShow
username          role
operator1         operator
USERID            admin
Bob               admin
```

sshKey

Adds, displays, and removes SSH public keys for individual users..

Syntax:

```
sshKey <cmd> [-u <username>] [<input>]
```

Options:

```
show
```

Displays the SSH public keys in the users authorized_keys file.

```
add "<key>"
```

Add key to the users authorized_keys file. Must be enclosed within double-quotes ("<key>").

```
rem <index>
```

Remove key at <index> for the user.

```
rem -all
```

Remove all keys for the usern.

```
-u <username>
```

Perform the operation on the user <username> (for admins only).

Sample Output:

```
-> sshKey show
 Index   Key
---------------------------------
    1      "ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEArNiSexu30rZjs1HAXbDBwTgJgcxLF..."
```

> **NOTES:** Adds, displays, and removes SSH public keys for individual users. Users with administrative privileges may use the -u option to manage keys for other users.

loginMode

Allows the user to change the user interface authentication requirements.

Syntax:

```
loginMode [mode]
```

Options:

```
[mode]
```

Determines how users are able to login.

Sample Output:

```
-> loginMode 1
Mode successfully changed to: 1 = Password is not required
```

NOTES: This command displays or changes how users are authenticated when connecting to the GUI or CLI. With no parameters, the current login mode is displayed. The login mode can be changed by specifying a single integer parameter indicating which of the following modes should become active:

0 - Username and password required

1 - Password is not required

2 - Username / password are not required

3 - LDAP Authentication

When usernames are disabled, all users will be logged on as the administrative user. There is no way to change this behavior.

setldapSvrIpAddr

Modify the LDAP Server IP Address.

Syntax:

```
setldapSvrIpAddr [ipaddress]
```

Options:

```
[ipaddress]
```

The IP address of the ldap server you wish to set to authenticate in the format "192.168.0.1"

Sample Output:

```
-> setldapSvrIpAddr 192.168.0.29
```

NOTES: This sets the LDAP server ip address. The LDAP server will be contacted for remote authentication.

setldapSvrPort

Modify the LDAP Server Port.

Syntax:

```
setldapSvrPort [port]
```

Options:

```
[port]
```

The port of the LDAP server to be authenticated.

Sample Output:

```
-> setldapSvrPort 389
```

NOTES: Sets the LDAP server port. The LDAP server will be contacted for remote authentication.

idleTimeoutGet

Retrieves the UI idle timeout value.

Syntax:

```
idleTimeoutGet [--all] [--cli] [--gui]
```

Options:

```
--all
```

Display all timeouts. This is the default if no options are specified.

```
--cli
```

Display the timeout for CLI sessions.

```
--gui
```

Display the timeout for GUI sessions.

Sample Output:

```
-> idleTimeoutGet
Timeout is 600 seconds.
```

NOTES: Displays the system idle timeouts for the GUI and CLI interfaces. If set to zero, the timeout is disabled.

idleTimeoutSet

Set the UI idle timeout value.

Syntax:

```
idleTimeoutSet [--all | --cli | --gui] seconds]
```

Options:

```
[seconds]
```
  Idle timeout value (in seconds).
```
--all
```
  Set the idle timeout for both the CLI and the GUI to the same value.
```
--cli
```
  Set the idle timeout for the CLI. This is the default if no identifier is specified.
```
--gui
```
  Set the idle timeout for the GUI.

Sample Output:

```
-> idleTimeoutSet --all 700
Timeout is set to 700 seconds.
```

> **NOTES:** Modifies the idle timeout for the CLI or GUI interfaces. Timeouts are in number of seconds, with zero meaning the timeout is disabled.

sessionTimeoutDisable

Disable the idle timeout for the current CLI session.

Syntax:

```
sessionTimeoutDisable
```

Options:

```
None.
```

Sample Output:

```
-> sessionTimeoutDisable
Disabled session idle timeout.
```

> **NOTES:** Disables the idle timeout for the current CLI session. This value does not persist across instances of the session (i.e., each time you log on, it will default back to the system default value).

sessionTimeoutEnable

Enable the idle timeout for the CLI session.

Syntax:

```
sessionTimeoutEnable
```

Options:

```
None.
```

Sample Output:

```
-> sessionTimeoutEnable
Enabled session idle timeout.
```

NOTES: Enables the idle timeout for the current CLI session. This value does not persist across instances of the session (i.e., with each log on, it defaults back to the system default value).

loginMsgGet

Displays the CLI login message for this device.

Syntax:

```
loginMsgGet
```

Options:

```
None.
```

Sample Output:

```
-> loginMsgGet
Welcome message: Be certain to logout when you are finished using the
CLI.
```

NOTES: Displays the current login welcome message for the CLI. This message can be customized with the loginMsgSet command.

loginMsgSet

Set the CLI login message for this device.

Syntax:

```
loginMsgSet [message] [-clear]
```

Options:

```
[message]
```
   The login message (should be encapsulated in quotes " ").

```
[-clear]
```
   Clear welcome message.

Sample Output:

```
-> loginMsgSet "Be certain to logout when you are finished using the
CLI."
Welcome message set successfully
```

NOTES: This command can be used to modify the welcome message displayed when logging onto the CLI. The current message can be viewed with the loginMsgGet command.

loginNameGet

Display the name of the device displayed when logged into via telnet.

Syntax:

```
loginNameGet
```

Options:

   None.

Sample Output:

```
-> loginNameGet
login-name: Switch1
```

NOTE: The login name is an arbitrary string displayed prior to a user attempting a login to a new CLI session. This command displays the current value of this string.

loginNameSet

Modify the string displayed to users prior to attempting a login for a new CLI session.

Syntax:

```
loginNameSet [option]
```

Options:

```
[name]
```
   Set the login name.
```
[-clear]
```
   Clear the login name.

Sample Output:

```
-> loginNameSet Switch1
login-name set successfully
```

> **NOTES:** The login name is an arbitrary string displayed prior to a user attempting a login to a new CLI session. This command modifies this string. The current name can be viewed with the loginNameGet command.

serialAuthGet

Displays the current serial authentication status.

Syntax:

```
serialAuthGet
```

Options:

   None.

Sample Output:

```
-> serialAuthGet
Serial authentication is currently disabled.
```

> **NOTES:** Displays whether user login and authentication is required on the serial console of the system.

serialAuthSet

Enable or disable authentication on the systems serial console.

Syntax:

```
serialAuthSet [enable]
```

Options:

```
[enable]
```
   Boolean value to enable or disable serial authentication.

Sample Output:

```
-> serialAuthSet 1
Serial authentication set to enabled.
```

> **NOTES:** This command is used to enable or disable user authentication on the serial console. Use 1 as the only parameter to enable authentication, or 0 to disable authentication. Exit and re-login for the setting to take effect. The login mode must be set to 0 or 1 for authentication to take place on the serial port.

uiConfig

View or configure the user interfaces to this device.

Syntax:

```
uiConfig [-telnet 0|1] [-https 0|1] [-http 0|1] [-ftp
0|1] [-snmp 0|1]
```

Options:

[telnet]

Enable/disable access to the device via telnet..

[https]

Enable or disable https access.

[http]

Enable or disable http access.

[ftp]

Enable or disable ftp access.

[snmp]

Enable or disable SNMP access.

Sample Output:

```
-> uiconfig -telnet 0
Successfully modified configuration.
 Option        Value
----------------------
 telnet        0
 http          1
 https         0
 ftp           1
 snmp          1
```

> **NOTES:** There are several access methods for this device. With no arguments, this command displays which access methods are available, and whether each method is enabled or disabled.

This command can also be used to enable or disable various access methods. Use '-<proto> 0' to disable a protocol, and '-<proto> 1' to enable. Multiple operations may be specified in a single command.

A reboot is not required for this command. Depending on the specific method, it may take 5-10 seconds for the changes. Note that existing connections (for example a telnet session) will not be affected by disabling the underlying access method.

Any access method supported by the device (but not listed here) cannot be configured and is always enabled.

genPost

Displays General POST results and runs tests

Syntax:

```
genPost
```

Options:

```
None.
```

Sample Output:

```
MasterMM202-> genpost

ID Gen POST Name                  Result   ENA DEM PWR Result-Ext
------------------------------------------------------------------
*  POST TEST CPU                  NO-RUN    Y   N   Y
*  POST TEST MEM DRAM             NO-RUN    Y   N   Y
*  POST TEST BOOTROM IMG CHKSUM   NO-RUN    Y   N   Y
*  POST TEST RUN-TIME IMG CHKSUM  NO-RUN    Y   N   Y
*  POST TEST JMP RAM              NO-RUN    Y   N   Y
6  POST TEST I2C PROBE            NO-RUN    Y   Y   N
7  POST TEST FAN TRAYS            NO-RUN    Y   Y   N
8  POST TEST PWR SUPPLIES         NO-RUN    Y   Y   N
9  POST TEST REAL-TIME CLOCK      NO-RUN    Y   Y   N
10 POST TEST PROTO                NO-RUN    Y   Y   N
```

NOTES: Displays General POST results and runs tests.

IpoIbConfigShow

Displays the IPoIB Enable/Disable setting.

Syntax:

```
IpoIbConfigShow
```

Options:

```
None.
```

Sample Output:

```
-> IpoIbConfigShow
```

IpoIbConfigEnable

Enable IPoIB feature.

Syntax:

```
IpoIbConfigEnable <- IpoIb enable>
```

Options:

```
None.
```

Sample Output:

```
-> IpoIbConfigEnable
```

IpoIbConfigDisable

Disables the IPoIB feature.

Syntax:

```
IpoIbConfigDisable
```

Options:

```
None.
```

Sample Output:

```
-> IpoIbConfigDisable
```

IpoIbAddressShow

Displays the IpoIb IP address and associated netmask.

Syntax:

```
IpoIbAddressShow
```

Options:

```
None.
```

Sample Output:

```
-> IpoIbAddressShow
```

> **NOTES:** IPoIB must be enabled to display these fields.

IpoIbAddressSet

Change the IpoIb IP address and associated network mask.

Syntax:

```
IpoIbAddressSet -h ipaddress -m netMask
```

Options:

```
[-h ipaddress]
```
the new IP address in dotted notation format 'xxx.xxx.xxx.xxx'.
```
[-m netMask]
```
network mask. (in hexidecimal format).

Sample Output:

```
-> IpoIbAddressSet -h 123.45.6.789
```

> **NOTES:** IPoIB must be enabled to display these fields.

IpoIbAddressSetIPv6

Changes the IPoIB IPv6 address and netmask.

Syntax:

```
IpoIbAddressSetIPv6 <ipv6Address/netMaskLen>
```

Options:

```
<ipv6Address/netMaskLen>  See NOTES.
```

Sample Output:

```
-> IpoIbAddressSetIPv6 <ipv6Address/netMaskLen>
```

> **NOTES:** The `<ipv6Address/netMaskLen>` is made up of two major components, separated by a '/'. The ipv6Address is in notation format 'a:b:c:d:e:f:g:h', where embedded 0s can be shortened (for example, 'a:b::g:h'). The netMaskLen is a value between 0 and 128 inclusive.

IpoIbAddressShowIPv6

Displays the IPoIB IPv6 Address and associated netmask settings.

Syntax:

```
IpoIbAddressShowIPv6
```

Options:

```
None.
```

Sample Output:

```
-> IpoIbAddressShowIPv6
```

exit

Exit the CLI.

Syntax:

```
exit
```

Options:

```
None.
```

Sample Output:

```
exit
```

Chassis

hwCheck

Runs a system check at the chassis level and returns current hardware status.

Syntax:

```
hwCheck
```

Options:

```
[n - 0=default, 1=verbose]
```

Sample Output:

```
-> hwCheck
Chassis hardware status: GOOD
```

> **NOTES:** Returns OK or provides detailed status/warning/error information. If an error/warning is detected, this command automatically provides verbose information.

hwMonitor

Displays current port states, fan speeds, temperatures, and voltages.

Syntax:

```
hwMonitor [slot] [onepass]
```

Options:

```
[slot]
```

Slot number to display. 0 for current slot of the management board.

```
[onepass]
```

Display the current values and exits. Do not continuosly monitor.

Sample Output:

```
(Sample 1)
-> hwMonitor 0
System monitor, Uptime: 0 days 4 hours 55 minutes 37 seconds
     MODULES          MODULES(cont)    MODULES(cont)        Power & Fans
  SLOT|P|TMP|VLT    SLOT|P|TMP|VLT   SLOT|P|TMP|VLT   SLOT|P|SPEED   SLOT|P|SPEED
     | |E|W|E|W        | |E|W|E|W       | |E|W|E|W       | |            | |
  M201 |*| | | |    L101 |*| | | |        | | | | |   P201| |        F201|*| 6840
  M202 |*| | | |    L102 |*| | | |        | | | | |   P202|*|        F202|*| 6480
  S105A|*| | | |    L103 |*| | | |        | | | | |   P203| |        F203|*| 6480
  S105B|*| | | |    L104 |*| | | |        | | | | |   P204|*|        F204|*| 6840
Max Temp=60C (Management Module 202)
Legend: P=pres/abs TMP=Temperature VLT=Voltage E=Err W=Warn (X=Err/Warn found)


(Sample 2)
-> hwMonitor 0
System monitor, Uptime: 0 days 4 hours 57 minutes 2 seconds
DETAIL FOR DEVICE:
   Leaf 101
Temperatures: Base Board:           [fusion]=31C            [asic]=40C
Voltages: Base Board
    [FPGA_1.5V_AUX]=1.555   [PCIE_1.5V_BULK]=1.480       [3.3V_AUX]=3.312
    [VDD_1.0V_BULK]=0.980   [AVTR_1.2V_BULK]=1.161    [AVDD_1.5V_BULK]=1.474
       [3.3V_BULK]=3.290     [AVTT_1.5_BULK]=1.485      [PLL_2.5_BULK]=2.391
                    InfiniBand Port Status
       L101P01[ Down ]          L101P02[ Down ]          L101P03[ Down ]
       L101P04[ Down ]          L101P05[ Down ]          L101P06[ Down ]
       L101P07[ Down ]          L101P08[ Down ]          L101P09[ Down ]
       L101P10[ Down ]          L101P11[ Down ]          L101P12[ Down ]
       L101P13[ Down ]          L101P14[ Down ]          L101P15[ Down ]
       L101P16[ Down ]          L101P17[ Down ]          L101P18[Active]
L101P19S105BP18[Active]  L101P20S105BP17[Active]  L101P21S105AP21[Active]
L101P22S105BP15[Active]  L101P23S105AP23[Active]  L101P24S105AP24[Active]
L101P25S105AP25[Active]  L101P26S105AP26[Active]  L101P27S105AP27[Active]
L101P28S105AP28[Active]  L101P29S105AP29[Active]  L101P30S105AP30[Active]
L101P31S105BP06[Active]  L101P32S105BP05[Active]  L101P33S105BP04[Active]
L101P34S105BP03[Active]  L101P35S105BP02[Active]  L101P36S105BP01[Active]
```

NOTES: To exit monitoring, press the enter key.

The slot number may require a prefix. The prefix for a Leaf is 'L', the prefix for a Spine is 'S', the prefix for a Mangement Module is 'M', the prefix for a Power Supply is 'P', the prefix for a Fan is 'F', and the prefix for the Chassis is 'C'. The Leaf, Spine and Management Module slots do not require a prefix. The Power Supply, Fan and Chassis slots require a prefix since their slot numbers overlap with the Leaf, Spine and Management Module slot numbers.

showIBNodeDesc

Displays the IB Node subnet management agent (SMA) Description..

Syntax:

```
showIBNodeDesc [-d]
```

Options:

```
[-d]
```
Show the default Node Name for this unit.

Sample Output:

```
-> showIBNodeDesc
IB Node (SMA) Description is = QLogic 12300 GUID=0x00066a00030001xa
```

NOTES: Entering this command with no parameters displays the current Node Description. The -d option will display the unit's default value.

setIBNodeDesc

Changes the IB Node (SMA) Description.

Syntax:

```
setIBNodeDesc "New Node String"
```

Options:

None.

Sample Output:

```
-> setIBNodeDesc "QLogic 12300 GUID=0x00066a00da000"
IB Node (SMA) Description successfully changed to = QLogic 12300
```

NOTES: New node name must be enclosed in quotes. Node name must be less than 64 characters.

setIBNodeDescFormat

Changes the IB Node (SMA) description format modifier.

Syntax:

```
setIBNodeDescFormat [format]
```

Options:

```
[format]
```
Use 0 or 1 to set the format

Sample Output:

```
-> setIBNodeDescFormat
Format = 0
```

> **NOTES:** If the format is not specified, then the current format selection value is displayed. When the format is 0, the SMA node description is in a verbose format. When the format is one the format is brief. The brief format is more consistent with the CLI/GUI Port Stat port naming.

fruInfo

Displays the EEProm contents.

Syntax:

```
fruInfo [slot|'-all']
```

Options:

```
[slot]
```
Slot that you wish to display EEProm contents. Defaults to 0 (chassis EEProm)

Sample Output:

```
-> fruInfo
xInfo_ChassisInfo:
      RecType:     1            LastRec:       0          LenMult:     0
      ReadOnly:    1            RecordFormat:  2          RecLen:     14
      LogicalLen:  020
      HdrChkSum:   c6
      ChassisGuid: 00066a000300012a
      SlotCount:   1        (IB Mods in Chassis)
      SlotNumbers: 81       (pairs:ext0|Slt1)
      CmeAccess:   80       (bits:ext0|Slt1|Cme)
      SlotNumber:   0 <-Record accessed via this slot
      CmeAccessBits: 2      (Access slot relative)
      ProxyAccess: 0        (Access slot relative)
      LockDrivesCTR: 0      (Clear to Remove interlock)
      MechLock:    1
      NodeCount:   0
...
```

> **NOTES:** The '-all' parameter performs this command against all available slots for the device.

chassisQuery

Displays information about modules inserted in the chassis that support firmware updates. This includes information regarding cardType, Board Support Package (BSP) and SlotNumber.

Syntax:

```
chassisQuery [-master] [-slave] [slot] [-showType]
[-type card_type] [-ignoreInvalidType]
```

Options:

```
[-master]
```
   Display the master slot number

```
[-slave]
```
   Display slave slot number

```
[n]
```
   slot number (numeric)

```
[-showType]
```
   display the card type

```
[-type card_type]
```
   display slots that have given card type

```
[-ignoreInvalidType]
```
   do not return an error if an invalid card type is supplied

Sample Output:

```
MasterMM202-> chassisQuery
slots: 201 202
```

> **NOTES:** Entering this command with no parameters will display all currently occupied card slots.

showInventory

Displays a detailed list of all hardware within the chassis.

Syntax:

```
showInventory
```

Options:

   None.

Sample Output:

```
MasterMM202-> showInventory
_____
            Leaf 101
----------------------------------
            GUID - 00066A00EC002712
  Manufacturer Id - 00066A
Manufacturer Name - SST
    Mfg Date/Time - 2009/04/09  10:00
            Model - 12800-LF18
      Part Number - 220052-004-B
     Product Name - 12800 QDR 18P Leaf Module
    Serial Number - USF1150920031
          Version - 004-B
...
```

setBeacon

Allows the user to view/set the chassis beacon.

Syntax:

```
setBeacon [beacon]
```

Options:

```
[beacon]
```
   1=on, 0=off.

Sample Output:

```
-> setBeacon 0
```

NOTES: Calling this function with no argument displays its current value.

setSystemContact

Allows the user to set/show the chassis system contact.

Syntax:

```
setSystemContact [contact]
```

Options:

```
[contact]
```
   System contact string.

Sample Output:

```
-> setSystemContact [contact string]
```

> **NOTES:** If the system contact is not specified, the current system contact is displayed.

setSystemName

Allows the user to set/show the chassis system name.

Syntax:

```
setSystemName [name]
```

Options:

```
[name]
```
System name string

Sample Output:

```
-> setSystemName [name string]
```

> **NOTES:** If the system name is not specified, the current system name is displayed.

setSystemLocation

Allows the user to set/show the chassis system location.

Syntax:

```
setSystemLocation [location]
```

Options:

```
[location]
```
System location string

Sample Output:

```
-> setSystemLocation [location string]
```

> **NOTES:** If the system location is not specified, the current system location is displayed.

Network

ifShow

Displays information for all the network interfaces for the switch, including the management port and the internal Ethernet network.

Syntax:

```
ifShow [ifName]
```

Options:

```
[ifName]
```

The network interface name.

Sample Output:

```
-> ifShow
sbe0 (unit number 0):
       Flags: (0x8063) UP BROADCAST MULTICAST ARP RUNNING
       Type: ETHERNET_CSMACD
       Internet address: 192.168.100.9
       Broadcast address: 192.168.111.255
       Internet address: 172.26.3.19
       Broadcast address: 172.26.15.255
       Netmask 0xfffff000 Subnetmask 0xfffff000
       Net 0xc0a86000 Subnet 0xc0a86000
       Ethernet address is 00:06:6a:01:99:c8
       Metric is 0
       Maximum Transfer Unit size is 1500
       2524301 octets received
       6604130 octets sent
       28728 packets received
       14310 packets sent
       19665 broadcast packets received
       31 broadcast packets sent
       0 multicast packets received
       0 multicast packets sent
```

NOTES: This routine displays the attached network interfaces for debugging and diagnostic purposes. If ifName is given, only the interfaces belonging to that group are displayed. If ifName is omitted, all attached interfaces are displayed.

routeShow

Displays the OOB LAN IP routes.

Syntax:

```
routeShow
```

Options:

```
None.
```

Sample Output:

```
-> routeShow

ROUTE NET TABLE
destination      gateway            flags  Refcnt  Use         Interface
----------------------------------------------------------------------------
0.0.0.0          172.26.0.1         3      2       18966       sbe0
0.0.0.0          172.26.3.35        3      0       0           sbe0
127.1.0.0        127.1.1.2          101    0       0           sbe1
127.2.0.0        127.2.0.1          101    0       0           sbe1
172.26.0.0       172.26.3.35        101    0       0           sbe0
172.26.0.0       172.26.3.19        101    -1      0           sbe0
----------------------------------------------------------------------------


ROUTE HOST TABLE
destination      gateway            flags  Refcnt  Use         Interface
----------------------------------------------------------------------------
127.0.0.1        127.0.0.1          5      3       14209       lo0
----------------------------------------------------------------------------
```

ping

Send ping packets to a specified host.

Syntax:

```
ping [ipAddress] [packetCount]
```

Options:

```
[IpAddress]
```

The IP address of the network host to ping.

```
[packetCount]
```

The number of packets with which to ping the host (default is 5).

Sample Output:

```
-> ping 172.26.0.254
PING 172.26.0.254: 56 data bytes
64 bytes from 172.26.0.254: icmp_seq=0. time=0. ms
64 bytes from 172.26.0.254: icmp_seq=1. time=0. ms
64 bytes from 172.26.0.254: icmp_seq=2. time=0. ms
64 bytes from 172.26.0.254: icmp_seq=3. time=0. ms
64 bytes from 172.26.0.254: icmp_seq=4. time=0. ms
----172.26.0.254 PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 0/0/0
```

NOTES: This routine spawns a process to send ping packets to the specified IP address. If packetCount is given, the process exits after that number of packets are sent. If packetCount is ommitted, a default the number of packets defaults to 5.

ping6

Sends ping packets to a specified IPv6 address.

Syntax:

```
ping6 [-n] [-q] [-v] [-c <packetCount>] [-I <interface>]
<ipv6Address>
```

Options:

[-n]

Numeric output only.

[-q]

Only display summary lines.

[-v]

Verbose output.

[-c <packetCount>]

The number of packets to ping the host with (default is 5).

[-I <interface>]

The local interface name to use (useful for link local ping).

[ipv6Address]

The network host to ping, scoped address accepted.

Sample Output:

```
-> ping6 ipv6Address
```

NOTES: This routine spawns a process to send ping packets to the specified IPv6 address. If packetCount is given, the process exits after that number of packets are sent. If packetCount is omitted, a default the number of packets defaults to 5.

showChassisIpAddr

Displays the chassis IP address.

Syntax:

```
showChassisIpAddr
```

Options:

```
None.
```

Sample Output:

```
-> showChassisIpAddr
Chassis IP Address: 172.26.0.221 Net mask: 255.255.240.0
```

setChassisIpAddr

Change the OOB LAN IP address and network mask.

Syntax:

```
setChassisIpAddr -h ipaddress -m netMask
```

Options:

```
-h ipaddress
```
   The new IP address in dotted notation format 'xxx.xxx.xxx.xxx'.
```
-m netMask
```
   The network mask. (may be in dotted notation or hexidecimal format)

Sample Output:

```
-> setChassisIpAddr -h 172.26.0.221 -m 255.255.240.0
```

NOTES: Changing the chassis IP address will drop the connection to the CLI if not using the console port and may cause the device to become unreachable.

setChassisIpv6Addr

Changes the chassis IPv6 address.

Syntax:

```
setChassisIpv6Addr [<ipv6Address> /<maskLen>]
```

Options:

```
<ipv6Address>
```

The new IPv6 address in notation format 'a:b:c:d:e:f:g:h', embedded 0s can be shortened, eg a:b::g:h.

```
<maskLen>
```

Network mask length. (0-128).

Sample Output:

```
-> setChassisIpv6Addr <ipv6Address>
```

> **NOTES:** Changing the chassis IPv6 address may drop the connection to the CLI if not using the console port, and may cause the device to become unreachable.

delChassisIpv6Addr

Deletes the chassis IPv6 address.

Syntax:

```
delChassisIpv6Addr <ipv6Address>
```

Options:

None.

Sample Output:

```
-> delChassisIpv6Addr <ipv6Address>
```

> **NOTES:** Changing the chassis IPv6 address may drop the connection to the CLI if not using the console port, and may cause the device to become unreachable. This command does not affect link-local or autoconfigured addresses.

showChassisIpv6Addr

Displays the chassis IPv6 address.

Syntax:

```
showChassisIpv6Addr
```

Options:

None.

Sample Output:

```
-> showChassisIpv6Addr
```

autoConfIPv6Enable

Enables IPv6 address autoconfiguration.

Syntax:

```
autoConfIPv6Enable
```

Options:

```
None.
```

Sample Output:

```
-> autoConfIPv6Enable
```

NOTES: This command allows prefix learning from attached routers, and does not affect link-local or manually configured addresses.

autoConfIPv6Disable

Disables IPv6 address autoconfiguration.

Syntax:

```
autoConfIPv6Disable
```

Options:

```
None.
```

Sample Output:

```
-> autoConfIPv6Disable
```

NOTES: Prevents prefix learning from attached routers, and does not affect link-local or manually configured addresses.

autoConfIPv6Show

Displays IPv6 address autoconfiguration setting(s).

Syntax:

```
autoConfIPv6Show
```

Options:

```
None.
```

Sample Output:

```
-> autoConfIPv6Show
```

ndpShow

Displays the IPv6 Neighbors table.

Syntax:

```
ndpShow
```

Options:

```
None.
```

Sample Output:

```
-> ndpShow
```

showDefaultRoute

Displays the default gateway IP address.

Syntax:

```
showDefaultRoute
```

Options:

```
None.
```

Sample Output:

```
-> showDefaultRoute
Gateway IP Address: 172.26.0.254
```

> **NOTES:** This is the IP address for the default gateway to route packets from the OOB mangement port to an external network.

setDefaultRoute

Change the default gateway IP address.

Syntax:

```
setDefaultRoute -h ipaddress
```

Options:

```
-h ipaddress
```
The default gateway IP address in dotted decimal format (xxx.xxx.xxx.xxx).

Sample Output:

```
setDefaultRoute -h 172.26.0.235
```

NOTES: This allows the user to configure the IP address for the default gateway to route packets from the OOB management port to an external network.

arpShow

Displays the contents of the ARP table.

Syntax:

```
arpShow
```

Options:

```
None.
```

Sample Output:

```
-> arpShow

LINK LEVEL ARP TABLE
destination     gateway            flags  Refcnt  Use          Interface
-------------------------------------------------------------------------
127.1.1.1       02:06:6a:ff:ff:fd   405    3      12121958     sbe1
127.2.0.2       02:06:6a:ff:ff:fd   c05    2      71421        sbe1
172.26.0.1      00:19:e8:a0:69:f9   405    1      0            sbe0
172.26.3.35     00:06:6a:01:99:c8   405    1      0            lo0
-------------------------------------------------------------------------
```

hostShow

Displays a list of remote hosts names, internet address and alias(es).

Syntax:

```
hostShow
```

Options:

None.

Sample Output:

```
-> hostShow
hostname        inet address        aliases
--------        ------------        -------
localhost       127.0.0.1           CHASSIS
slot1           127.1.0.1
slot2           127.1.0.2
slot3           127.1.0.3
slot4           127.1.0.4
slot5           127.1.0.5
slot6           127.1.0.6
slot7           127.1.0.7
slot8           127.1.0.8
slot9           127.1.0.9
slot10          127.1.0.10
slot11          127.1.0.11
slot12          127.1.0.12
slot13          127.1.0.13
slot14          127.1.0.14
slot15          127.1.0.15
slot16          127.1.0.16
slot17          127.1.0.17
slot18          127.1.0.18
slot19          127.1.0.19
slot20          127.1.0.20
slot21          127.1.0.21
slot22          127.1.0.22
slot23          127.1.0.23
slot24          127.1.0.24
slot25          127.1.0.25
slot26          127.1.0.26
slot27          127.1.0.27
slot28          127.1.0.28
slot29          127.1.0.29
slot30          127.1.0.30
slot31          127.1.0.31
slot32          127.1.0.32
slot33          127.1.0.33
slot34          127.1.0.34
slot35          127.1.0.35
slot36          127.1.0.36
mm1             127.1.1.1           switchA
mm2             127.1.1.2           switchB
Master          127.2.0.1
Slave           127.2.0.2
WRS-Template    172.26.3.35
home            10.32.2.150
```

dhcpEnable

Enable the DHCP client for this system.

Syntax:

```
dhcpEnable
```

Options:

```
None.
```

Sample Output:

```
MasterMM202-> dhcpEnable
DHCP is enabled
```

> **NOTES:** Enables the DHCP client subsystem, requests a DHCP lease, and then configures the interface with the lease data from the server,

dhcpDisable

Disable the DHCP client and release any acquired IP address

Syntax:

```
dhcpDisable
```

Options:

```
None.
```

Sample Output:

```
MasterMM202-> dhcpDisable
DHCP is disabled
```

> **NOTES:** Any DHCP-acquired IP address will be released. The interface will then be configured to the default static values from the current bootline.

dhcpShow

Display DHCP client information.

Syntax:

```
dhcpShow [-v]
```

Options:

```
-v verbose output
```

Sample Output:

```
MasterMM202-> dhcpShow
Client state = BOUND
Assigned IP address: 172.26.3.35
Client subnet mask: 255.255.240.0
DHCP server: 172.26.1.20
Default IP router: 172.26.0.1
Client lease duration: 518400 secs (421299 remaining)
```

dnsParamsShow

Displays the stored domain named server (DNS) parameters.

Syntax:

```
dnsParamsShow
```

Options:

```
None.
```

Sample Output:

```
MasterMM202-> dnsParamsShow
DNS Resolver       : Enabled
DNS Server Address : xxx.xxx.xxx.xxx
Local Domain Name  : st.qlogic.com
```

> *NOTE:*
>
> This command retrieves the stored configuration parameters used for domain name resolution.

dnsParamsSet

Changes the DNS configuration parameters.

Syntax:

```
dnsParamsSet [-s <ipaddress>] [-d <domain name>] [-e <1
or 0>]
```

Options:

```
-s <ipaddress>
```
   DNS server IP address in dotted notation format (xxx.xxx.xxx.xxx)

```
-d <domain name>
```
   The local domain name where this switch is installed (limit 32 chars).

```
-e <1 or 0>
```
   1 = enable the DNS resolver, 0 = disable the DNS resolver.

Sample Output:

```
dnsParamsSet -e 0

MasterMM202-> dnsParamsShow
DNS Resolver       : Disabled
DNS Server Address : xxx.xxx.xxx.xxx
Local Domain Name  : st.qlogic.com
```

NOTE:
> The DNS resolver cannot be enabled until both the server address and local domain name have been configured. It is necessary to manually reboot the switch in order to start or stop the DNS resolver.

Firmware

fwUpdate (12300)

Updates the firmware.

Syntax:

```
fwUpdate [host user password dir filename]
```

Options:

```
host
```
The name of the host where the firmware file resides.

```
user
```
The FTP user name.

```
password
```
The FTP user password.

```
dir
```
After logging in, the directory to **cd** to.

```
filename
```
The name of the firmware file.

Sample Output:

```
-> fwUpdate
Enter 1 for FTP, 2 for local file: 1
Ftp Server IP Address:[192.168.0.195]
Ftp username:[ftp] xxxxx
Ftp password:[ftp] xxxxx
File Directory:[PATH TO FIRMWARE FILE]
File name:[xxx.pkg]
```

NOTES: Firmware update works by using ftp to retrieve the firmware file, then writes the file to flash. Omitting any of the options causes the user to be prompted for the information.

fwUpdateSlot

Updates the firmware on a specific line card.

Syntax:

```
fwUpdateSlot slotnumber [host user password dir
filename]
```

Options:

```
slotnumber
```
The chassis slot numberto update.

```
[host]
```
The name of the host where the firmware file resides.

```
[user]
```
The FTP user name.

```
[password]
```
The FTP user password.

```
[dir]
```
After logging in, the directory to **cd** to.

```
[filename]
```
The name of the firmware file.

Sample Output:

```
-> fwUpdateSlot 1
Enter 1 for FTP, 2 for local file: 1
Ftp Server IP Address:[192.168.0.195]
Ftp username:[ftp] xxxxx
Ftp password:[ftp] xxxxx
File Directory:[PATH TO FIRMWARE FILE]
File name:[xxxx.pkg]
```

NOTES: Firmware update works by using ftp to retrieve the firmware file, then writes the file to flash. Omitting any of the options causes the user to be prompted for the information.

fwUpdateChassis

Updates the firmware for all cards in a chassis of a particular type, or everything.

Syntax:

```
fwUpdateChassis (all,management,evic, fvic) [noprompt]
[reboot]
```

Options:

```
(all,management,evic, fvic)
```
Type of card to update. See notes for details.

```
[noprompt]
```
Will not prompt the user for ftp information (uses the saved values).

```
[reboot]
```
Upon successful completion, reboots the updated cards .

Sample Output:

```
-> fwUpdateChassis all reboot
```

NOTES: Firmware update works by retrieving the firmware file from an ftp server. The file is saved locally and then written to flash. Using the 'all' option, all cards in the chassis will be updated. This assumes that all firmware files are in the same location. The user will be prompted for the names of each file for each line card type present. The system can save the values entered as defaults for future firmware updates.

fwListFiles

Lists the contents of the firmware directory.

Syntax:

```
fwListFiles
```

Options:

```
None.
```

Sample Output:

```
-> fwListFiles

Listing Directory /firmware:
[PATH TO FIRMWARE FILE] / [FIRMWARE FILE NAME]
```

NOTES: The firmware directory temporarily stores firmware files before they are written to flash.

fwShowUpdateParams

Display the default update firmware settings.

Syntax:

```
fwShowUpdateParams
```

Options:

```
None.
```

Sample Output:

```
-> fwShowUpdateParams
Firmware update ftp configuration settings:
 host:[0.0.0.0]
 user:[ftp]
 password:[ftp]
 directory:[]
 filename management:[]
 filename evic:[]
 filename fvic:[]
```

fwSetUpdateParams

Change the default update firmware settings.

Syntax:

```
fwSetUpdateParams -c cardtype [-h hostname] [-u
username] [-p password] [-d directory] [-f filename]
```

Options:

```
-c - cardtype
```
Choices are: 'management', 'fvic', 'evic'
```
-h - hostname
```
The host name or IP address of the FTP server.
```
-u - username
```
The name of the user accessing the ftp server.
```
-p - password
```
The password of the user accessing the ftp server.
```
-d - directory
```
The directory containing the firmware file.
```
-f - filename
```
The firmware file name.

> **NOTES:** Modifies the default update firmware parameters. Except for the filename option, all parameters are shared across all card types. The cardtype parameter is only required when specifying the filename.

showCapability

Display capability and feature information for a specific release.

Syntax:

```
showCapability [-key feature]
```

Options:

```
[-key feature]
```
Displays information for a particular feature.

Sample Output:

```
-> showCapability
fwPush: 1
```

showLastScpRetCode

Display the return code from the last SCP Firmware Push.

Syntax:

```
showLastScpRetCode slot | -all
```

Options:

```
slot
```
The slot number in the chassis.
```
-all
```
All slots in the chassis.

Sample Output:

```
-> showLastScpRetCode 101
SCP: Slot 101 Last Exit Code: 0: Success
```

> **NOTES:** This allows for automated systems to determine if a SCP firmware push was successful or not.

fwVersion

Displays the firmware versions for a unit.

Syntax:

```
fwVersion [slot]
```

Options:

```
slot
```
Slot number.

Sample Output:

```
Management Module 201 Information -----------------
  Firmware Version: 5.0.0.0.73
  Firmware build:   5_0_0_0_73
  Firmware BSP:     xt3
  MBC Version:      None
  Bootrom Version:  5.0.0.0.57
```

bootQuery

Displays boot image version information.

Syntax:

```
bootQuery [slot] [-active | -alternate | -all]
```

Options:

```
[slot]
```
Slot number.

```
[-active]
```
Displays the version of the active firmware image.

```
[-alternate]
```
Displays t]he version of the alternate firmware image.

```
[-all
```
Displays the versions for the primary and alternate firmware images.

Sample Output:

```
MasterMM202-> bootQuery 201
Primary firmware version: 5.0.0.0.73
```

bootSelect

Select the next boot image to be used.

Syntax:

```
bootSelect [slot] [-i index] [-alternate] [-version
version] [-noprompt]
```

Options:

```
[slot]
```
The slot number using the next boot image.

```
[-i index]
```
The index of the boot image to be used next.

```
[-alternate]
```
Chooses the alternate image to be used next.

```
[-version version]
```
Chooses a specific version to be the image to be used next.

```
[-noprompt]
```
Displays the current configuration only.

Sample Output:

```
MasterMM202-> bootSelect 201
Currently installed firmware versions
index : alias    : version
------------------------
*# 1   : image1   : 5.0.0.0.73;
   2   : image2   : 5.0.0.0.72;

  * - indicates Primary image (will run at next reboot)
  # - indicates Active image

Default boot image index = 1
...
```

NOTES: This command allows the user to set the next boot image for the device. A '*' next to the image entry indicates the currently selected bootimage. A '#' indicates the currently active boot image.

Subnet Management

smControl

Starts and stops the embedded subnet manager.

Syntax:

```
smControl start | stop | restart | status
```

Options:

```
Start
```
Start the SM.
```
Stop
```
Stop the SM.
```
restart
```
Restarts the SM.
```
status
```
Prints out the SM Status.

Sample Output:

```
-> smControl start
Starting the SM...
```

smConfig

Configure startup parameters of the embedded subnet manager.

Syntax:

```
smConfig [query] [startAtBoot yes|no] [startOnSlaveCmu yes|no]
```

Options:

```
query
```
Display present settings, no change
```
startAtBoot
```
Start the subnet manager at chassis boot
```
startOnSlaveCmu
```
Start subnet manager on the slave CMU

Sample Output 1:

```
-> smConfig
Start at boot? [Y]
Start on slave CMU? [N]
```

Sample Output 2:

```
-> smconfig startAtBoot yes startOnSlaveCmu yes
Saving....
Saving complete...
```

NOTES: Use this command to configure the subnet manager. Changes to these parameters do not take effect until the chassis management card(s) is rebooted. This command is only available on the master chassis management card.

smShowLids

Displays the LID of the subnet manager.

Syntax:

```
smShowLids
```

Options:

```
None.
```

Sample Output:

```
sm_state = MASTER   count = 572781   LMC = 0, Topology Pass count = 339, Priority = 0, Mkey = 0x0

-----------------------------------------------------------------------------------------
QLogic 12300 GUID=0x00066a00da000000 172.26.2.2 MM 1, Ch
-----------------------------------------------------------------------------------------
Node[  0] => 00066a000600013c  (2)  ports=24, path=
       Port ---- GUID  ---- (S)  LID      LMC     _VL_   __MTU__  _WIDTH_  ____SPEED____   CAP_MASK   N#  P#
          0 00066a000600013c  4  LID=0001 LMC=0000  8  8   2k  2k   4X  4X  2.5   2.5   00000a4a    0   0
          4 0000000000000000  4                     8  8   2k  2k   4X  4X  2.5/5 5.0   00000000    1
22  4 22
          5 0000000000000000  4                     8  8   2k  2k   4X  4X  2.5/5 5.0   00000000    1
23  4 23
          6 0000000000000000  4                     8  8   2k  2k   4X  4X  2.5/5 5.0   00000000    1
24  4 24
          7 0000000000000000  4                     8  8   2k  2k   4X  4X  2.5/5 5.0   00000000    2
24  7 24
          8 0000000000000000  4                     8  8   2k  2k   4X  4X  2.5/5 5.0   00000000    2
23  7 23
          9 0000000000000000  4                     8  8   2k  2k   4X  4X  2.5/5 5.0   00000000    2
22  7 22
         22 0000000000000000  4                     8  8   2k  2k   4X  4X  2.5/5 5.0   00000000    3
16 22 16
         23 0000000000000000  4                     8  8   2k  2k   4X  4X  2.5/5 5.0   00000000    3
18 22 18
         24 0000000000000000  4                     8  8   2k  2k   4X  4X  2.5/5 5.0   00000000    3
17 22 17

...
```

NOTES: Use this command to display the current LID of the subnet manager.

smPriority

Display the priority of the embedded subnet manager.

Syntax:

```
smPriority
```

Options:

```
None.
```

Sample Output 1:

```
-> smPriority
The SM Priority is 0
```

> **NOTES:** The priority of the Subnet Manager (SM) determines which subnet manager will become the master SM for the fabric. Zero is the lowest priority and fifteen is the highest. The SM with the highest priority becomes the master SM for the fabric. The elevated priority value determines what the priority of the SM will be if it becomes the Master. This allows persistent failovers that do not automatically fail back by configuring the elevated priority to be higher than all normal priorities. This feature is effectively disabled when set to the default of zero. The subnet manager must be running to display the priority.

smSweepRate

Display or dynamically set the sweep rate of the subnet manager.

Syntax:

```
smSweepRate [sweepRate]
```

Options:

```
sweepRate
```

The sweep rate (in seconds) of the subnet manager. Valid values are 3-86400, or 0 to turn the sweep off. The sweepRate is the interval between the end of one sweep and the start of the next sweep.

Sample Output:

```
-> smSweepRate 300
The SM sweep rate has been set to 300 seconds
```

NOTES: The sweep rate determines how often the subnet manager scans the fabric for changes and events. The subnet manager must be running to display or dynamically set the sweep rate of the embedded subnet manager. Changes to sweepRate made with the smSweepRate command will only affect the subnet manager that is currently running and will not be persistent.

smForceSweep

Force a fabric sweep by the embedded subnet manager.

Syntax:

```
smForceSweep
```

Options:

```
None.
```

Sample Output:

```
-> smForceSweep
```

NOTES: Use this command to force a sweep by the subnet manager.

bmForceSweep

Force a fabric sweep by the embedded baseboard manager.

Syntax:

```
bmForceSweep
```

Options:

```
None.
```

Sample Output:

```
-> bmForceSweep
```

NOTES: Use this command to force a sweep by the baseboard manager.

smShowGroups

Display multicast group information in the embedded subnet manager.

Syntax:

```
smShowGroups [-h]
```

Options:

```
-h
```
Display the host name as part of the output.

Sample Output:

```
-> smShowGroups
Multicast Groups:
  join state key: F=Full N=Non S=SendOnly Member

0xff12601bffff0000:00000001ffffd5bb (c001)
  qKey = 0x00000000  pKey = 0xFFFF  mtu = 4  rate = 3  life = 19  sl = 0
  0x0011750000ffd5bb F

0xff12401bffff0000:00000000ffffffff (c000)
  qKey = 0x00000000  pKey = 0xFFFF  mtu = 4  rate = 3  life = 19  sl = 0
  0x00066a01a0007116 F   0x0002c902003fffd5 F   0x00066a00a00001ac F
  0x00066a01a000015d F   0x00066a00a00001a3 F   0x00066a00a00001dc F
  0x00066a00a000035a F   0x0011750000ffd5c2 F   0x0011750000ffd664 F
  0x0011750000ffd9c2 F   0x0011750000ffd9f8 F   0x0011750000ffd5b9 F
  0x0011750000ffda4a F   0x0011750000ffd5bb F   0x0011750000ffd9de F
```

NOTES: Use this command to display multicast group information in the subnet manager. This command is not available unless the subnet manager is in the Master mode.

smShowServices

Display subnet administration service records of the subnet manager.

Syntax:

```
smShowServices
```

Options:

None.

Sample Output:

```
-> smShowServices
******************************************************************
                 There is 1 Service Records
*******************************************************
Service ID          = 0x1100D03C34834444
Service GID         = 0xFE80000000000000:00066A000600013C
Service P_Key       = 0x0000
Service Lease       = infinite
Service Key         =
0x00 0x00 0x00 0x00  0x00 0x00 0x00 0x00  0x00 0x00 0x00 0x00  0x00
0x00 0x00 0x00
Service Name        = QLogic Fabric Executive service Rev 1.1
Service Data 8      =
0x00 0x00 0x00 0x00  0x00 0x00 0x00 0x00  0x00 0x00 0x00 0x00  0x00
0x00 0x00 0x00
Service Data 16     =
0x0000 0x0000  0x0000 0x0000  0x0000 0x0000  0x0000 0x0000
Service Data 32     =
0x0000 0x0000 0x0000 0x0000
Service Data 64     =
0x0000000000000000 0x0000000000000000
Service Expire Time = 0x0100000000000000
```

NOTES: The components(fields) of each service record are displayed. Each service record is stored in a location identified by a 'Slot' number which is displayed before any component of that Service Record. If a group of slots do not contain Service Records, the first slot of the empty group is displayed as 'empty'. This command is not available unless the subnet manager is in the Master mode.

smShowSubscriptions

Display event forwarding (subscription) table in the embedded subnet manager.

Syntax:

```
smShowSubscriptions
```

Options:

```
None.
```

Sample Output:

```
-> smShowSubscriptions
*****************************************************************
                There are 2 subscriptions
*****************************************************************
Subscriber GID        = 0xFE80000000000000:00066A00D8000163
Subscriber LID        = 0x0071
Subscriber PKey       = 0xFFFF
Subscriber Start LID  = 0x0001
Subscriber End LID    = 0xBFFF
Subscriber Record ID  = 0x00000001
Subscriber Inform Info =
GID                   = 0x0000000000000000:0000000000000000
Start LID             = 0xFFFF
End LID               = 0x0000
Is Generic?           = Yes
Subscribe?            = Subscribe
Type                  = All Types
Trap Number           = 0x0040
Queue Pair Number     = 0x000001
Response Time Value   = 19
Producer Type         = Subnet Management
*****************************************************************
Subscriber GID        = 0xFE80000000000000:00066A01A0007116
Subscriber LID        = 0x0007
Subscriber PKey       = 0xFFFF
Subscriber Start LID  = 0x0001
Subscriber End LID    = 0xBFFF
Subscriber Record ID  = 0x00000036
Subscriber Inform Info =
GID                   = 0x0000000000000000:0000000000000000
Start LID             = 0xFFFF
End LID               = 0x0000
Is Generic?           = Yes
Subscribe?            = Subscribe
Type                  = All Types
Trap Number           = 0x0043
Queue Pair Number     = 0x000001
Response Time Value   = 18
Producer Type         = Channel Adapter
*****************************************************************
                There are 2 subscriptions
```

NOTES: Use this command to display the event forwarding (subscription) table in the subnet manager. This command is not available unless the subnet manager is in the Master mode.

smMasterLMC

Display the LMC value of the subnet manager to be used on CA ports.

Syntax:

```
smMasterLMC
```

Options:

```
None.
```

Sample Output:

```
-> smMasterLMC
```

> **NOTES:** The value of the LMC determines how many LID's are assigned to an endport; 2 LMC LIDs are assigned to endports based on this value. For example, setting the LMC to a value of 3 will assign 8 LIDs per endport. The allowable LMC values are between 0 and 7, inclusive. The subnet manager must be running as MASTER to display the LMC value.

smShowMasterLid

Display the LID of the subnet manager.

Syntax:

```
smShowMasterLid
```

Options:

```
None.
```

Sample Output:

```
-> smShowMasterLid
The SM LID is 0x0001
```

> **NOTE:** Use this command to display the LID of the master subnet manager.

smShowLidMap

Display the LID-to-port GUID map for the subnet manager.

Syntax:

```
smShowLidMap
```

Options:

```
None.
```

Sample Output:

```
-> smShowLidMap
--------------------------------------------------------------------------------
SM is currently in the MASTER state, with Topology Pass count = 341
--------------------------------------------------------------------------------
Lid 0x0001: guid = 0x00066a000600013c, pass = 341, QLogic 12300 GUID=0x00066a00da000100
172.26.2.2 Spine 1, Ch
Lid 0x0002: guid = 0x00066a0007000170, pass = 341, QLogic 12300 GUID=0x00066a00da000100
172.26.2.2 Leaf 4, Chi
Lid 0x0003: guid = 0x00066a100600013c, pass = 341, QLogic 12300 GUID=0x00066a00da000100
172.26.2.2 Spine 1, Ch
Lid 0x0006: guid = 0x00066a00a0000248, pass = 229
Lid 0x0007: guid = 0x00066a01a0007116, pass = 341, st149
Lid 0x0008: guid = 0x0000000000000000, pass = 0
Lid 0x0028: guid = 0x0000000000000000, pass = 0
Lid 0x002a: guid = 0x0000000000000000, pass = 0
```

> **NOTES:** Use this command to display the LID-to-port GUID map of the subnet
> manager. This command is not available unless the subnet manager is in the Master
> mode.

smShowMaxLid

Display the highest LID allocated by the subnet manager.

Syntax:

```
smShowMaxLid
```

Options:

```
None.
```

Sample Output:

```
-> smShowMaxLid
The maximum LID is 0x0138
```

> **NOTES:** Use this command to display the highest LID allocated by the subnet
> manager. This command is not available unless the subnet manager is in the Master
> mode.

smSwitchLifetime

Allows the user to display/dynamically set the default switch lifetime of the subnet manager.

Syntax:

```
smSwitchLifetime [lifetime}
```

Options:

lifetime

The packet lifetime value between 0 and 31, inclusive.

Sample Output:

```
-> smSwitchLifetime 15
SM switch packet lifetime: 15 (~134217 microseconds)
```

> **NOTES:** The switch lifetime value determines the maximum time a packet may remain within a switch, calculated using the formula: 4.096 * (2 switchlifetime) in microseconds. The packet lifetime values may be between 0 and 31, inclusive. The subnet manager must be running to display or dynamically set the default switch lifetime. Changes will only affect the SM that is currently running and will not be persistent.

smHoqLife

Display or dynamically set the head of queue packet lifetime for switch ports.

Syntax:

```
smHoqLife [lifetime]
```

Options:

lifetime

The packet lifetime value between 0 and 31, inclusive.

Sample Output:

```
-> smHoqLife 9
SM HOQ Liftime: 9 (~2097 microseconds)
```

> **NOTES:** Use this command to display the maximum lifetime that a packet may remain at the head of a virtual lane transmission queue before it is discarded by a switch. Packet lifetime values are between 0 and 31, inclusive. These are calculated using the formula: 4.096 * (2 ^ switchlifetime) in microseconds. The subnet manager must be running to display, or to dynamically set the Hoq lifetime. Changes to Hoq lifetime set with the smHoqLife command only affect the currently running SM and will not be persistent.

smVLStall

Allows the user to display/dynamically set the VL stall value of the subnet manager.

Syntax:

```
smVLStall [packets]
```

Options:

```
packets
```
The number of sequential packets dropped before port enters VL stalled state.

Sample Output:

```
-> smVLStall 5
SM VL Stall Threshhold: 5 packets
```

> **NOTES:** Use this command to set the VL stall value for ports in the fabric. This value determines the how quickly a virtual lane for a particular switch or endport enters a 'stalled' state after dropping packets. The subnet manager must be running to display or to dynamically set VLStall. Changes to VLStall with the smVLStall command only affect the currently-running subnet manager and will not be persistent.

smShowSMParms

Display subnet manager parameters switch lifetime, HOQ lifetime, VLStall val, pkt lifetime, and dynamic PLT.

Syntax:

```
smShowSMParms
```

Options:

None.

Sample Output:

```
-> smShowSMParms
SM priority is set to 4
SM LMC is set to 0
SM sweep rate is set to 300
SM max retries on receive set to 3
SM max receive wait interval set to 250 millisecs
switchLifetime set to 15
HoqLife set to 9
VL Stall set to 5
packetLifetime constant is set to 18
Dynamic PLT ON using values: 1 hop=16, 2 hops=17, 3 hops=17, 4 hops=18,
5 hops=18, 6 hops=18, 7 hops=18, 8+hops=19
SM DBSync interval set to 900
SM topology errors threshold set to 0, max retry to 3
```

NOTES: Use this command to display a sampling of subnet manager parameters.

smPKeys

Configure a partition key (PKey) in the PKey table.

Syntax:

```
smPKeys
```

Options:

```
None.
```

Sample Output:

```
-> smPKeys
```

NOTES: PKeys are used for partitioning the subnet. Only configure PKeys if the host driver supports this. Invalid configuration of the PKey may render the fabric inoperable. The subnet manager must be running to display PKeys.

smInfoKey

Set the subnet manager key (SMInfo) value.

Syntax:

```
smInfoKey
```

Options:

```
None.
```

Sample Output:

```
-> smInfoKey
SM Key: 0x0000000000000001 (1 decimal).
```

> **NOTES:** Use this command to set the SM key. SM must be offline and key value is up to 8 byte hex. The subnet manager must be running to display the SMInfo key value.

smMgmtKey

Set the subnet manager management key (portInfo) value.

Syntax:

```
smMgmtKey
```

Options:

```
None.
```

Sample Output:

```
-> smMgmtKey 0x11
SM management key: 0x0000000000000011 (17 decimal)
```

> **NOTES:** Use this command to set the SM management key. SM must be offline and mkey value is up to 8 byte hex. The subnet manager must be running to display the SMInfo key value.

smOptionConfig

Use this command to configure support for non-default modes of operation.

Syntax:

```
smOptionConfig [clear | default | [def-mcgrp-create]
[dyn-plt]]
```

Options:

```
clear
```

Clears all set options.

```
default
```

Enable the default set of options (i.e., def-mcgrp-create and dyn-pit).

```
def-mcgrp-create
```

Auto create default multicast group. If using only this option, dyn-pit will be disabled.

```
dyn-pit
```

Enable dynamic packet lifetimes for pathrecord queries. If using only this option, def-mcgrp-create will be disabled.

Sample Output:

```
-> smOptionConfig dyn-plt
      [dyn-plt] Dynamic packet lifetime support is enabled
```

NOTES: The subnet manager must be running to display the non-default modes of operation.

smDefBcGroup

Displays the default multicast group configuration.

Syntax:

```
smDefBcGroup
```

Options:

```
None.
```

Sample Output:

```
-> smDefBcGroup
```

NOTES: The subnet manager must be running to display default multicast group information.

- Valid MTU values are 1(256), 2(512), 3(1024), 4(2048), and 5(4096)

- Valid RATE values are 2(2.5GB), 3(10GB), 4(30GB), 5(5GB), 6(20GB), 7(40GB), 8(60GB), 9(80GB), 10(120GB)

- Valid Values for SL are 0 to 0xF

- Valid Values for QKEY are 0 to 0xFFFFFFFF

- Valid Values for FlowLabel are 0 to 0xFFFFF

■ Valid Values for TClass are 0 to 0xff

smGidPrefix

Set the Subnet Prefix (default=0xfe80000000000000).

Syntax:

```
smGidPrefix [prefix]
```

Options:

```
None.
```

Sample Output:

```
-> smGidPrefix
Subnet Prefix: 0xfe80000000000001
```

> **NOTES:** Use this command to set the subnet prefix of the SM. The SM must be offline and the subnet prefix value is 8 byte hex. The subnet manager must be running to display default multicast group information.

smSubnetSize

Set/display the subnet size for the subnet manager.

Syntax:

```
smSetSubnetSize
```

Options:

```
None.
```

Sample Output:

```
-> smSubnetSize
```

> **NOTES:** Use this command to tune the SM to handle the configured fabric size. The subnet manager must be running to display subnet size information.

smTopoErrorThresh

Set/display the error threshold for a topology sweep.

Syntax:

```
smTopoErrorThresh
```

Options:

```
None.
```

Sample Output:

```
-> smTopoErrorThresh 100
Set topology error threshold to:100
```

NOTES: Sets the maximum number of errors the SM may encounter during a sweep before abandoning the sweep. The subnet manager must be running to display the error threshold for a topology sweep.

smTopoAbandonThresh

Set/display the max consecutive times the SM can abandon a sweep due to too many errors.

Syntax:

```
smTopoAbandonThresh
```

Options:

```
None.
```

Sample Output:

```
-> smTopoAbandonThresh 3
Set topology sweep abandonment threshold to:3
```

NOTES: The subnet manager must be running to display this information.

smMaxRetries

Displays the maximum number of subnet manager receive retries

Syntax:

```
smMaxRetries
```

Options:

```
None.
```

Sample Output:

```
-> smMaxRetries 3
Set max retries to:3

-> smMaxRetries
Max retries: 3
```

NOTES: The subnet manager must be running to display this information.

smRcvWaitTime

Displays the maximum time (in milliseconds) to wait for a reply to a subnet manager packet.

Syntax:

```
smRcvWaitTime
```

Options:

```
None.
```

Sample Output:

```
-> smRcvWaitTime 250
Set recieve wait time to: 250 milliseconds

-> smRcvWaitTime
Recieve wait time: 250 milliseconds
```

NOTES: The subnet manager must be running to display this information.

smNonRespDropTime

Displays the seconds to wait before dropping a non-responsive node.

Syntax:

```
smNonRespDropTime
```

Options:

```
None.
```

Sample Output:

```
-> smNonRespDropTime 300
Set Nonresponsive node drop time to: 300 seconds

-> smNonRespDropTime
Non-responsive node drop time: 300 seconds
```

NOTES: The subnet manager must be running to display this information.

smNonRespDropSweeps

Displays the sweeps to wait before dropping a non-responsive node.

Syntax:

```
smNonRespDropSweeps
```

Options:

```
None.
```

Sample Output:

```
-> smNonRespDropSweeps 3
Set Nonresponsive node drop sweeps to: 3 sweeps

-> smNonRespDropSweeps
Non-responsive node drop sweeps: 3 sweeps
```

NOTES: The subnet manager must be running to display this information.

smLogLevel

Displays or dynamically sets the Subnet Manager logging level.

Syntax:

```
smLogLevel [<loglevel>]
```

Options:

```
loglevel
    logging level 1-7
```

Sample Output:

```
-> smLogLevel 4
```

NOTES:
Levels: 1=WARN+, 2=INFINI_INFO+, 3=INFO+, 4=VERBOSE+, 5=DEBUG2+, 6=DEBUG3+, 7=TRACE+.

The subnet manager (SM) must be running to use this command. Changes made with this command will only affect the currently running SM, and will be forgotten if the FM is restarted or the chassis is rebooted. To make persistent changes edit the FM XML configuration file.

smLogMode

Displays or dynamically sets the Subnet Manager logging mode.

Syntax:

```
smLogMode [<logmode>]
```

Options:

```
logmode
```
  logging mode 0 or 1

Sample Output:

```
-> smLogMode 1
```

> **NOTES:**
> 0 - use normal logging levels
>
> 1 - logging is quieted by downgrading the majority of fatal, error, warn and infiniinfo log messages to level
>
> 3 (INFO) and only outputting user actionable events when LogLevel is 1 or 2.
>
> The Subnet manager must be running to use this command. Changes made with this command will only affect the currently running SM, and will will be forgotten if the FM is restarted or the chassis is rebooted. To make persistent changes edit the FM XML config file.

smLogMask

Displays or dynamically sets the Subnet Manager logging mask for a specific subsystem.

Syntax:

```
smLogMask <subsystem> [<mask>]
```

Options:

```
subsystem
```
  The Fabric Manager (FM) subsystem.
```
mask
```
  The bit mask needed for logging to enable.

Sample Output:

```
-> smLogMask SM
```

NOTES:

Subsystems: CS, MAI, CAL, DVR, IF3, SM, SA, PM, PA, BM, FE, APP

The Subnet manager must be running to use this command. Changes made with this command will only affect the currently running SM, and will will be forgotten if the FM is restarted or the chassis is rebooted. To make persistent changes edit the FM XML config file.

smMcLidTableCap

Displays the limit of the multicast LIDs that are available for allocation.

Syntax:

```
smMcLidTableCap
```

Options:

```
None.
```

Sample Output:

```
-> smMcLidTableCap 1024
Set mc lid limit to 1024

-> smMcLidTableCap
Mc lid limit: 1024
```

NOTES: Setting this value to zero disables limiting multicast LIDs. The subnet manager must be running to display this information.

smMasterPingInterval

Displays the ping interval (in seconds) of the subnet manager.

Syntax:

```
smMasterPingInterval
```

Options:

```
None.
```

Sample Output:

```
-> smMasterPingInterval 4
Set master ping interval to 3 seconds

-> smMasterPingInterval
Master ping interval: 4 seconds
```

> **NOTES:** Value must be between 3 and 10. The subnet manager must be running to display this information.

smMasterPingFailures

Displays the number of allowable master ping failures.

Syntax:

```
smMasterPingFailures
```

Options:

```
None.
```

Sample Output:

```
-> smMasterPingFailures 3
Set master ping failures to 3 failures

-> smMasterPingFailures
Master ping failures: 3 failures
```

> **NOTES:** Value must be between 2 and 5. The subnet manager must be running to display this information.

smDbSyncInterval

Displays how often a Master subnet manager should perform a full synchronization with the standby subnet manager(s).

Syntax:

```
smDbSyncInterval
```

Options:

None.

Sample Output:

```
-> smDbSyncInterval
SM DB full sync interval currently set to 15 minutes

-----SM DB SYNCHRONIZATION interval set to 900 seconds, 2 SM(s) in fabric-----
MASTER SM node at QLogic 12300 GUID=0x00066a00d90003fa, LID 0x0008, PortGuid
0x00066a00d90003fa
     Sync Capability is  SUPPORTED
STANDBY SM node at st44, LID 0x0100, PortGuid 0x00066a00a0000357
     Sync Capability is  SUPPORTED
     Full sync status is      SYNCHRONIZED
     Time of last Full sync is THU APR 10 15:37:47 2008
     Time of last INFORM records sync is THU APR 10 15:37:47 2008
     Time of last GROUP records sync is THU APR 10 15:37:47 2008
     Time of last SERVICE records sync is THU APR 10 15:37:47 2008
```

NOTES: Value must be between 0 and 60 minutes (0=OFF). The subnet manager must be running to display this information.

smDynamicPlt

Set/display dynamic packet lifetime values.

Syntax:

```
smDynamicPlt [index] [plt]
```

Options:

```
index
```

The index of the entry in the table to be changed. Range is 1-9..

```
plt
```

The packet lifetime value. The range is 0-63.

Sample Output:

```
-> smDynamicPlt
Index: 1 PLT Value: 16 (~268435 usec)
Index: 2 PLT Value: 17 (~536870 usec)
Index: 3 PLT Value: 17 (~536870 usec)
Index: 4 PLT Value: 18 (~1073741 usec)
Index: 5 PLT Value: 18 (~1073741 usec)
Index: 6 PLT Value: 18 (~1073741 usec)
Index: 7 PLT Value: 18 (~1073741 usec)
Index: 8 PLT Value: 19 (~2147483 usec)
Index: 9 PLT Value: 19 (~2147483 usec)

Dynamic packet lifetime values for pathrecord queries are enabled
(use the smOptionConfig command to change)



-> smDynamicPlt 1 15
Index: 1 PLT Value: 15 (~134217 usec)
Index: 2 PLT Value: 17 (~536870 usec)
Index: 3 PLT Value: 17 (~536870 usec)
Index: 4 PLT Value: 18 (~1073741 usec)
Index: 5 PLT Value: 18 (~1073741 usec)
Index: 6 PLT Value: 18 (~1073741 usec)
Index: 7 PLT Value: 18 (~1073741 usec)
Index: 8 PLT Value: 19 (~2147483 usec)
Index: 9 PLT Value: 19 (~2147483 usec)

Dynamic packet lifetime values for pathrecord queries are enabled
(use the smOptionConfig command to change)
```

NOTES: Setting values to numbers greater than 19 give an effectively-infinite packet lifetime. The subnet manager must be running to display dynamic packet lifetime values.

sm1xLinkMode

Displays how the subnet manager handles 1x links.

Syntax:

```
sm1xLinkMode
```

Options:

```
None.
```

Sample Output:

```
-> sm1xLinkMode
Mode is 'off'. Erroneous 1x links will be activated normally.
-> sm1xLinkMode ignore
-> sm1xLinkMode
Mode is 'ignore'. Erroneous 1x links will not be activated.
```

NOTES: When set to 'off', all links come up normally. When set to 'ignore', links that only come up at 1x (when they were enabled for a higher rate) are forced down. These downed ports can be queried to aid debugging errors in the fabric. The subnet manager must be running to display this information.

smTrapThreshold

Display the port auto-disable urgent trap threshold value (in minutes).

Syntax:

```
smTrapThreshold
```

Options:

```
None.
```

Sample Output:

```
-> smTrapThreshold
Trap Threshold is 0 (disabled).

-> smTrapThreshold 20

-> smTrapThreshold
Trap Threshold is 20 traps/minute.
```

NOTES: When enabled, ports generating urgent traps at a rate higher than the threshold will be disabled. This value can range from 10 to 100 traps/minute. Set

the value to 0 to disable this feature. The subnet manager must be running to display this information.

smAppearanceMsgThresh

Displays the threshold for appearance and disappearance messages.

Syntax:

```
smAppearanceMsgThresh
```

Options:

```
None.
```

Sample Output:

```
-> smAppearanceMsgThresh
Message Threshold is 0 (disabled).

-> smAppearanceMsgThresh 3

-> smAppearanceMsgThresh
Message Threshold is 3 messages/sweep.
```

> **NOTES:** This command sets the threshold for the number of fabric appearance and disappearance log messages that may be logged as NOTICEs per sweep by the SM. A value of zero causes all such messages to be logged at the NOTICE level. A value greater than zero will cause the priority of any subsequent messages to be logged at the INFO priority. The subnet manager must be running to display this information.

smPmBmStart

Allows the user to set/display whether performance manager and baseboard manager starts along with the subnet manager.

Syntax:

```
smPmBmStart [enable] [disable]
```

Options:

```
enable
```

Enable the start of the PM and BM at SM start-up.

```
disable
```

Disable the start of the PM and BM at SM start-up.

Sample Output:

```
-> smPmBmStart
SM is enabled
PM is enabled
BM is enabled
FE is enabled

-> smPmBmStart disable
SM is enabled
PM is disabled
BM is disabled
FE is enabled
```

smShowRemovedPorts

Displays the ports that have been automatically removed from the fabric..

Syntax:

```
smShowRemovedPorts
```

Options:

```
None.
```

Sample Output:

```
-> smShowRemovedPorts
Disabled Ports:
 Node 0x00066a0007000001: QLgoic 12300 GUID=0x00066a00da000001 Leaf 1,
Chip A
    Port 12: Trap Threshold Exceeded
    Port 23: 1x Link Width
```

> **NOTES:** This displays ports that have been removed from the fabric automatically by the SM, such as when a 1x link mode is set to 'ignore' or when a port has exceeded its urgent trap threshold. This command is not available unless the subnet manager is in Master mode.

smShowCounters

Display various statistics and counters maintained by the subnet manager.

Syntax:

```
smShowCounters
```

Options:

```
None.
```

Sample Output:

```
-> smShowCounters
COUNTER: THIS SWEEP LAST SWEEP      TOTAL
----------------------------------- ---------- ---------- ----------
   SM State transition to DISCOVERY:        0          0          2
      SM State transition to MASTER:        0          0          1
     SM State transition to STANDBY:        0          0          1
    SM State transition to INACTIVE:        0          0          0
       Total transmitted SMA Packets:     123        711       2181
            Direct Routed SMA Packets:     123        711       2122
               LID Routed SMA Packets:       0          0         40
                  SMA Query Retransmits:       0          0         18
         SMA Query Retransmits Exhausted:       0          0          3
                   SM TX GET(Notice):        0          0          0
                   SM TX SET(Notice):        0          0          0
                  SM RX TRAP(Notice):        0          0          0
            SM TX TRAPREPRESS(Notice):        0          0          0
            SM TX GET(NodeDescription):       0        148        444
                 SM TX GET(NodeInfo):        0        148        444
               SM TX GET(SwitchInfo):        0          6         18
...
```

NOTES: This command is not available unless the subnet manager is in Master mode.

smResetCounters

Resets various statistics and counters maintained by the subnet manager.

Syntax:

```
smResetCounters
```

Options:

```
None.
```

Sample Output:

```
-> smResetCounters
```

pmShowCounters

Displays various statistics and counters about the performance manager (PM).

Syntax:

```
pmShowCounters
```

Options:

```
None.
```

Sample Output:

```
-> pmShowCounters
```

pmResetCounters

Resets various statistics and counters about the performance manager (PM).

Syntax:

```
pmResetCounters
```

Options:

```
None.
```

Sample Output:

```
-> pmResetCounters
```

pmShowRunningTotals

Displays Running Total Counters for the performance manager (PM) for all ports in a fabric.

Syntax:

```
pmShowRunningTotals
```

Options:

```
None.
```

Sample Output:

```
-> pmShowRunningTotals
```

bmPriority

Display the priority of the embedded baseboard manager.

Syntax:

```
bmPriority
```

Options:

```
None.
```

Sample Output:

```
-> bmPriority
The BM Priority is 0
The BM Elevated Priority is disabled
```

> **NOTES:** The priority of the Baseboard Manager (BM) determines which BM will become the master BM for the fabric. The priority values are 0 (lowest) and 15 (highest). The BM with the highest priority becomes the master BM for the fabric. The elevated priority value determines what the priority of the BM will be if it becomes Master. This allows persistent failovers that do not automatically fail back by configuring the elevated priority to be higher than all normal priorities. This feature is effectively disabled when set to the default of zero. The subnet manager must be running to display the priority.

pmPriority

Display the priority of the embedded performane manager.

Syntax:

```
pmPriority
```

Options:

```
None.
```

Sample Output:

```
-> pmPriority
The PM Priority is 0
The PM Elevated Priority is disabled
```

> **NOTES:** The priority of the Performance Manager(PM) determines which PM will become the master PM for the fabric. The priority values are 0 (lowest) and 15 (highest). The PM with the highest priority becomes the master PM for the fabric. The elevated priority value determines what the priority of the PM will be if it becomes Master. This allows persistent failovers that do not automatically fail back by configuring the elevated priority to be higher than all normal priorities. This feature is effectively disabled when set to the default of zero. The subnet manager must be running to display the priority.

smRestorePriority

Restore normal priorities from elevated states for the SM, BM, and PM.

Syntax:

```
smRestorePriority [sm | bm | pm | all]
```

Options:

```
sm
```
   Restore normal SM priority
```
bm
```
   Restore normal BM priority
```
pm
```
   Restore normal PM priority
```
all
```
   Restore normal priorities for the SM, BM and PM.

Sample Output:

```
-> smRestorePriority
```

> **NOTES:** This command restores the normal priorities of various subnet managers after they have elevated their priority following a failover. Issuing this command allows the 'unsticking' of a sticky failover. Issuing this command without arguments will restore the normal priorities of the SM, BM, and PM.

smResetConfig

Resets the XML configuration for the embedded subnet manager (ESM) to factory defaults. This command is only available on the Master chassis management card.

Syntax:

```
smResetConfig [-noprompt]
```

Options:

```
-noprompt
```
   proceed with a configuration reset without a confirmation prompt.

Sample Output:

```
-> smResetConfig
Proceed with configuration reset? [N] y
Default XML configuration file has been generated.
```

smShowConfig

Displays the XML configuration file.

Syntax:

```
smShowConfig [-infoOnly | -contentOnly] [-noprompt]
```

Options:

```
-infoOnly
```

Displays the timestamp for the XML configuration file.

```
-contentOnly
```

Displays the contents of the XML configuration file.

```
-noprompt
```

Do not prompt to 'Continue' for each page of displayed output.

Sample Output:

```
-> smShowConfig
<!-- Common FM configuration, applies to all FM instances/subnets -->
<Common>
  <!-- Various sets of Applications which may be used in Virtual Fabrics -->
  <!-- Applications defined here are available for use in all FM instances. -->
  <!-- Additional Applications may be defined here or per FM instance. -->
  <!-- Applications specified per FM instance will add to -->
  <!-- instead of replace these Application definitions. -->
  <Applications>
    <!-- Each Application can have one or more ServiceIDs and/or MGIDs. -->
    <!-- These will be matched against PathRecord and Multicast SA queries -->
    <!-- so that the returned SLID/DLID, PKey, SL, etc are appropriate for -->
    <!-- the Virtual Fabric(VF) which contains the application(s). -->

    <!-- Every Application must have a unique <Name> -->
    <!-- The name must be unique among all Application names within an -->
    <!-- FM instance. -->
    <!-- When defined at Common level must be unique within all instances. -->
    <!-- The name is limited to 64 characters and is case sensitive. -->

    <!-- ServiceIDs are 64 bit values which identify applications within the -->
    <!-- PathRecord query and are typically used within the InfiniBand -->
    <!-- Communication Manager (CM) protocol to identify the application for -->
Continue? [Y]
...
```

NOTES: With no arguments, the XML configuration file timestamp and contents will be displayed, one screen at a time. Enter **Y** or **Enter** at the prompt to continue displaying command output. Enter **N** at the prompt to terminate the output. The **-infoOnly** and **-contentOnly** flags will limit the information displayed. Use the

-noprompt flag to send all output to the screen at once.This command is only available on the Master Chassis management card.

Log

logShow

Displays the log file.

Syntax:

```
logShow
```

Options:

```
None.
```

Sample Output:

```
-> logshow

W|2006/10/04 20:26:31.176U: Thread "Log" (0x8fdab3b0)
        Log: Unable to Send Trap: 523:Bad:65535
W|2006/10/04 20:26:31.176U: Thread "CPU1" (0x1)
        Fcpi: Target Device 1 (2e7 p1) NPort Id 0x0106d1: Connection Restored
W|2006/10/04 20:26:31.196U: Thread "Log" (0x8fdab3b0)
        Log: Unable to Send Trap: 523:Bad:65535
W|2006/10/04 20:26:31.216U: Thread "CPU1" (0x1)
        Fcpi: Target Device 2 (b31 p2) NPort Id 0x0106d2: Connection Restored
W|2006/10/04 20:26:31.236U: Thread "CPU1" (0x1)
        Fcpi: Target Device 3 (d94 p3) NPort Id 0x0106d3: Connection Restored
W|2006/10/04 20:26:31.246U: Thread "Log" (0x8fdab3b0)
        Log: Unable to Send Trap: 523:Bad:65535
W|2006/10/04 20:26:31.256U: Thread "CPU1" (0x1)
        Fcpi: Target Device 6 (ac6 p6) NPort Id 0x0106d6: Connection Restored
W|2006/10/04 20:26:31.276U: Thread "Log" (0x8fdab3b0)
        Log: Unable to Send Trap: 523:Bad:65535
W|2006/10/04 20:26:31.326U: Thread "Log" (0x8fdab3b0)
        Log: Unable to Send Trap: 523:Bad:65535
W|2006/10/04 20:28:29.912U: Thread "tTelnetd" (0x8fe143e0)
        Osa: telnetd: connection requested by 192.168.0.107
W|2006/10/04 20:46:26.113U: Thread "tTelnetd" (0x8fe143e0)
        Osa: telnetd: connection requested by 192.168.0.107
W|2006/10/05 19:37:08.727U: Thread "tTelnetd" (0x8fe143e0)
        Osa: telnetd: connection requested by 192.168.0.46
W|2006/10/05 20:19:20.101U: Thread "tTelnetd" (0x8fe143e0)
```

NOTES: This command displays the log file that is contained in RAM.

logClear

Clears the log file.

Syntax:

```
logClear [-noprompt]
```

Options:

```
-noprompt
```
   delete all log messages without prompting the user

Sample Output:

```
-> logClear
Ram Log cleared
```

NOTES: This command deletes all entries in the log file.

logConfigure

Configures the log settings.

Syntax:

```
logConfigure
```

Options:

   None.

Sample Output:

```
-> logConfigure
Type Q or X to exit.
Please enter the number corresponding to what you want to configure.
index : name           : description
-------------------------
  1   : Device         : Logging device. (IE. Ram, syslog, etc)
  2   : Preset         : General log filter.

Select: 1
Configurable devices
index : name      : |D|F|E|A|W|P|C|I|P|N|1|2|3|4|5|
--------------------------------------------------
  1   : Ram       : |X|X|X|X|X| | | |X| | | | | |
  3   : Console   : |X|X|X|X|X| | | |X| | | | | |
  5   : Syslog    : |X|X|X|X|X| | | |X| | | | | |
Type Q or X to exit

Enter the device index you wish to configure: 1
Level: Dump [1]
Level: Fatal [1]
Level: Error [1]
Level: Alarm [1]
Level: Warning [1]
Level: Partial [0]
Level: Config [0]
Level: Info [0]
Level: Periodic [0]
Level: Notice [1]
Level: Debug1 [0]
Level: Debug2 [0]
Level: Debug3 [0]
Level: Debug4 [0]
Level: Debug5 [0]
Log device configuration changed
...
```

NOTES: This is an interactive command to configure log settings. This involves setting the log levels that are active.

Definitions:

- Preset: Enable or disable each log level that may be generated on the system. The log presets act as a general filter. For example, if the **Info** logging level is disabled in the presets, no **Info** messages will be shown on any output device, regardless of **Info** log setting for a device.

- Device: Enable a device to display or process log messages of each level. The level must also be active in Preset for log messages to be processed for a device.

- Syslog: Configure the syslog host ip address and port.

logResetToDefaults

Restores the log file default settings.

Syntax:

```
logResetToDefaults [-noprompt]
```

Options:

```
-noprompt
```
   Restore the defaults without prompting the user.

Sample Output:

```
-> logResetToDefaults
Log configuration has been reset
```

logSyslogConfig

Configure the syslog host IP address.

Syntax:

```
logSyslogConfig [-h xxx.xxx.xxx.xxx] [-p xxxx] [-f xx]
[-m x]
```

Options:

```
-h ip_address
```
   Sets the host IP address in dotted decimal format (xxx.xxx.xxx.xxx).

```
-p port
```
   The host port number on which the syslog server is listening.

```
-f facility
```
   The syslog facility to use in the messages.

```
-m mode
```
   Determines whether the syslog is to be put into a special OEM mode.

Sample Output:

```
-> logSyslogConfig -h 172.26.0.202
Successfully configured the syslog host
```

> **NOTES:** If configured the device can forward its log messages to a syslog host. This command allows a user to configure the host and port to send messages to and the facility to use in the messages. The facility value assignments are:

- 0 kern

- 1 user

- 2 mail

- 3 daemon

- 4 auth

- 5 syslog

- 6 lpr

- 7 news

- 8 uucp

- 9 cron

- 10 authpriv

- 11 ftp

- 12 ntp

- 13 audit

- 14 alert

- 15 clock

- 16 local0

- 17 local1

- 18 local2

- 19 local3

- 20 local4

- 21 local5

- 22 local6

- 23 local7

Additional configuration may be necessary to fully configure the log system.

logShowConfig

Display the current log configuration settings.

Syntax:

```
logShowConfig
```

Options:

```
None.
```

Sample Output:

```
Log Configuration for Slot 101:
-----------------------------
Configurable devices
index : name     : |D|F|E|A|W|P|C|I|P|N|1|2|3|4|5|
----------------------------------------------------
  1   : Ram      : |X|X|X|X|X| | | | |X| | | | | |
  3   : Console  : |X|X|X|X|X| | | | |X| | | | | |
  5   : Syslog   : |X|X|X|X|X| | | | |X| | | | | |

Configurable presets
index : name     : state
-----------------------------
  1   : Dump     : Enabled
  2   : Fatal    : Enabled
  3   : Error    : Enabled
  4   : Alarm    : Enabled
  5   : Warning  : Enabled
  6   : Partial  : Disabled
  7   : Config   : Disabled
  8   : Info     : Disabled
  9   : Periodic : Disabled
 15   : Notice   : Enabled
 10   : Debug1   : Disabled
 11   : Debug2   : Disabled
 12   : Debug3   : Disabled
 13   : Debug4   : Disabled
 14   : Debug5   : Disabled
```

logSyslogTest

Test the Syslog configuration.

Syntax:

```
logSyslogTest [severityType]
```

Options:

-e

Send Error severity CSM test message to Syslog.

-w

Send Warning severity CSM test message to Syslog

-n

Send Warning severity CSM test message to Syslog.

Sample Output:

```
-> logSyslogTest -e
Currently configured Syslog host is: 0.0.0.0 port 514 facility 22
Syslog configuration has been tested
```

NOTES: This command tests the Syslog configuration by sending CSM message(s) to registered Syslog servers.

Key Management

showKeys

Display the license keys for the unit.

Syntax:

```
showKeys
```

Options:

```
None.
```

Sample Output:

```
-> showKeys
-------------------------------------------------------
Key number:  1
Key:         XX7P91-2V9H6F-946XS3-3SCEV5-XXMZ5R-S
Description: Subnet Manager License
Status:      Active
```

> **NOTES:** License keys unlock various product software features.

addKey

Add a license key.

Syntax:

```
addKey [key]
```

Options:

```
key
```
  The license key to add.

Sample Output:

```
-> addKey XXxxx-YYYYY-ZZZZZ-11111-222222-3
```

removeKey

Remove a license key.

Syntax:

```
removeKey
```

Options:

None.

Sample Output:

```
-> removeKey
----------------------------------------------------
Key number:  1
Key:         XX7P91-XXXH6F-XXXQS3-YYCEV5-YYYZ5R-S
Description: Subnet Manager License
Status:      Active

Please enter the key number you wish to remove:
```

NOTES: After entering this command a list of available keys will be displayed. The keys are shown along with an associated number. At the prompt, enter the number of the key you wish to remove.

IbSwitchInfo

ismPortStats

Displays link error information associated with each switch port. These statistics include errors, dropped packets, discarded packets, and invalid packets.

Syntax:

```
ismPortStats [-clear] [-noprompt] [-cols X] [-port X]
[-leaf X] [-spine X]
```

Options:

```
-clear
```

Clears the statistics. Statistics are displayed first, then cleared.

```
-noprompt
```

Does not give the user a 'Continue' prompt for each page of display.

```
-cols X
```

Sets the number of columns to be displayed per line.

```
-port X
```

Specifies a port(s) to display.

```
-leaf X (12800 only)
```

Displays all ports for a specific leaf.

```
-spine X (12800 only)
```

Displays all ports for a specific spine.

Sample Output:

```
-> ismPortStats
Name        HSSM_BAY-1 HSSM_BAY-2
PhysState       Up       Up
PortState       Act      Act
LinkWidth       4X       4X
LinkSpeed   2.5Gbps 2.5Gbps
SymblErrors      0        0
ErrRecovery      0        0
LinkDowned       0        0
RcvErrors        0        0
RmtPhysErr       0        0
TxDiscards       0        0
InPKeyViol       0        0
OutPKeyViol      0        0
InRawViol        0        0
OutRawViol       0        0
LLIntegrity      0        0
ExcesBufOvr      0        0
```

NOTES: Port statistic descriptions:

- PhysState   - The physical state of the port.

- PortState   - The state of the link on this port.

- LinkWidth   - The currently active link width on this port.

- LinkSpeed   - The currently active link speed on this port. Port speed is LinkWidth * LinkSpeed.

- SymblErrors - The number of times a 8B10B encoding violation, or a disparity violation was detected. If multiple errors are detected simultaneously (in more than one lane), the counter only increments by one.

- ErrRecovery - The number of times the link error recovery proccess happened successfully.

- LinkDowned - The number of times the link error recovery process failed.

- RcvErrors   - Number of errors received on the port.

- RmtPhysErr - Number of remote physical errors received on the port.

- TxDiscards - Number of port transmit discards.

- InPKeyViol - Number of times PKey inbound invalid.

- OutPKeyViol - Number of times PKey outbound invalid.

- InRawViol   - Number of times raw inbound packet discarded.

- OutRawViol - Number of times raw outbound packet was discarded.

- LLIntegrity - Number of local link integrity errors.

- ExcesBufOvr - Number of excessive buffer overrun errors.

■ The options -leaf and -spine are only available on platforms with removable leaf and spine modules.

ismPortCounters

Displays a table comparision of transmit, receive and error counters cooresponding to each port of the module.

Syntax:

```
ismPortCounters [-clear] [-active] [-errors]
[-potential] [-stats] [-noprompt]
```

Options:

```
-clear
```

Clears the counters. Counters are first displayed, then cleared.

```
-active
```

Displays only the counters for ports in the active state.

```
-errors
```

Displays only the counters for ports with receive symbol errors.

```
-potential
```

Displays only the counters for ports with active link or width under their maximum supported value.

```
-stats
```

Displays the optional link error counters associated with each switch port.

```
-noprompt
```

Does not give the user a 'Continue' prompt for each page of display on switches with more than 24 ports.

Sample Output:

```
-> ismPortCounters -errors
No ports with symbol errors.
```

NOTES: Port counter descriptions:

- Transmit, Packets - The number of packets transmitted by the port, not including flow control packets.

- Transmit, Words - The number of data words transmitted by the port, not including flow control and VCRC data.

- Transmit, Discard - The number of transmit packets discarded by the port due to congestion or errors.

- Receive, Packets - The number of data packets received by the port, not including flow control packets.

- Receive, Words - The number of data words received by the port, not including flow control and VCRC data.

- Symbol, Errors - Number of symbol errors received on the port.

- Active, Speed - The active link speed of the port.

- Active, Width - The active link width of the port.

- Active, Potential - The port utilization based on the maximum supported link speed and maximum supported link width.

Optional Statistics

- PhysState - The physical state of the port.

- PortState - The state of the link on this port.

- LinkWidth - The currently active link width on this port.

- LinkSpeed - The currently active link speed on this port.

- SymbIErrors - The number of 8B10B encoding or disparity violations.

- ErrRecovery - The number of link error recovery proccess successes.

- LinkDowned - The number of link error recovery process failures.

- RcvErrors - Number of errors received on the port.

- RmtPhysErr - Number of remote physical errors received on the port.

- TxDiscards - Number of port transmit discards.

- InPKeyViol - Number of times PKey inbound invalid.

- OutPKeyViol - Number of times PKey outbound invalid.

- InRawViol - Number of times raw inbound packet discarded.

- OutRawViol - Number of times raw outbound packet was discarded.

- LLIntegrity - Number of local link integrity errors.

- ExcesBufOvr - Number of excessive buffer overrun errors.

- Note: The user is prompted to continue the output after each group of 24 ports are displayed. This command is best displayed with a terminal width of at least 120 columns.

ismLinearFwdb

Displays the entries in the linear forwarding table. LIDs and a cooresponding port are shown. A packet addressed to a LID will be forwared to the cooresonding port listed in the displayed table.

Syntax:

```
ismLinearFwdb [switch]
```

Options:

switch

switch number.

Sample Output:

```
-> ismLinearFwdb
Switch Leaf 101 Linear Fwdb (LFTTOP = 0x27):
 LID :: Port
0001      18 (L101P18)
0003      22 (L101P22S105BP15)
0004      21 (L101P21S105AP21)
0005      21 (L101P21S105AP21)
0006      21 (L101P21S105AP21)
0009      19 (L101P19S105BP18)
0014      0 ( )
001a      21 (L101P21S105AP21)
0027      20 (L101P20S105BP17)


Switch Leaf 102 Linear Fwdb (LFTTOP = 0x27):
 LID :: Port
0001      19 (L102P19S105BP36)
0003      21 (L102P21S105BP34)
0004      25 (L102P25S105AP07)
0005      21 (L102P21S105BP34)
0006      21 (L102P21S105BP34)
0009      20 (L102P20S105BP35)
0014      21 (L102P21S105BP34)
001a      0 ( )
0027      15 (L102P15)
...
```

NOTES: The 'set' option is not available on the SilverStorm 9000 products.

ismMultiFwdb

Displays the Multicast Forwarding database for the switch.

Syntax:

```
ismMultiFwdb [switch]
```

Options:

```
switch
```

Switch identifier.

Sample Output:

```
-> ismMultiFwdb
Switch Leaf 101 Multicast Fwdb:
c000     18 (L101P18) 19 (L101P19S105BP18) 21 (L101P21S105AP21)
c001     18 (L101P18) 19 (L101P19S105BP18) 21 (L101P21S105AP21)
c002     19 (L101P19S105BP18) 21 (L101P21S105AP21)
c003     18 (L101P18) 19 (L101P19S105BP18) 21 (L101P21S105AP21)
c004     19 (L101P19S105BP18) 21 (L101P21S105AP21)
c005     19 (L101P19S105BP18) 21 (L101P21S105AP21)
c006     19 (L101P19S105BP18) 21 (L101P21S105AP21)
c007     19 (L101P19S105BP18) 21 (L101P21S105AP21)
c008     19 (L101P19S105BP18) 21 (L101P21S105AP21)
c009     19 (L101P19S105BP18) 21 (L101P21S105AP21)
c00a     18 (L101P18) 19 (L101P19S105BP18) 21 (L101P21S105AP21)
c00b     18 (L101P18) 19 (L101P19S105BP18) 21 (L101P21S105AP21)
c00c     18 (L101P18) 19 (L101P19S105BP18) 21 (L101P21S105AP21)
...
```

> **NOTES:** This command is best displayed with a terminal width of at least 120 columns.

ismAutoClearConf

This feature clears the InfiniBand port statistic counters that have reached maximum.

Syntax:

```
ismAutoClearConf [disable | enable [warn] [log_first]]
```

Options:

enable

Enable the auto-clear feature.

disable

Disable the auto-clear feature.

warn

Generate warning log messages instead of the default information messages.

log_first

Log first clear (otherwise the first clear is not logged).

Sample Output:

```
-> ismAutoClearConf enable
Auto clear is enabled
```

NOTE: This feature will log every time a counter has reached its maximum capacity. This may be useful for diagnostics purposes, specifically for bad cables. This feature is only available on certain switch hardware platforms.

ismPortSet12x

Allows the user to view, set and unset port link width for specific port(s).

Syntax:

```
ismPortSet12x portName [enable8xBit] [-bounce]
```

Options:

portName

A valid 6 or 7 character port name (e.g., S1AL01a or L01P01).

enable8xBit

2=enable(8X only) 1=enable(AUTO 1X/4X/8X), 0=disable.

-bounce

Brings the active links down and back up if a new value is set.

Sample Output:

```
-> ismPortSet12x S1AL01a
S1AL01a 12x mode is DISABLED
```

NOTES: This command is included for backwards compatibility. The ismPortSetWidth command is preferred. This command ONLY works on platforms

supporting 8X link aggregation. NOTE: The option portName is case sensitive. To force links to only operate at 8X, use the '8X only' option.

Calling this command without the enable8xBit option will display the current values of the port(s).

Calling this function with the [-bounce] option disruptively brings active links down and back up so the links use the new setting immediately. Otherwise the new setting will be used the next time links retrain.

ismChassisSet12x

Allows the user to view, set and unset the chassis link width (including external ports).

Syntax:

```
ismChassisSet12x [enable8xBit] [-bounce]
```

Options:

```
enable8xBit
```
   2=enable(8X only) 1=enable(AUTO 1X/4X/8X), 0=disable.
```
-bounce
```
   Brings the active links down and back up if a new value is set.

Sample Output:

```
-> ismChassisSet12x
L01P01  8x mode is DISABLED
L01P02  8x mode is DISABLED
L01P03  8x mode is DISABLED
L01P04  8x mode is DISABLED
L01P05  8x mode is DISABLED
...
```

NOTES: This command included for backwards compatibility. The ismChassisSetWidth command is prefered.

This command ONLY works on switches supporting 8X link aggregation. This command will enable 8X link aggregation so that Internal AND External switching will be at 8x. Each 8X port is 2 aggregated 4x ports (i.e., a DUO), where one will be reported as an 8x port, and the other port is reported as disabled. Each of the two aggregated 4x ports must be connected to another enabled 8X Duo to establish a 8X link. To force links to only operate at 8X, use the '8X only' option. Calling this command without the enable8xBit will display the ports' current values.
Calling this function with the [-bounce] option disruptively brings active links down and back up so the links use the new setting immediately. Otherwise the new setting will be used the next time links retrain.

ismChassisSetSpeed

Allows the user to view and set port link speeds for the entire chassis.

Syntax:

```
ismChassisSetSpeed [speed] [-bounce]
```

Options:

```
speed
```

1=SDR 2=DDR 3=SDR/DDR 4=QDR 5=SDR/QDR 6=DDR/QDR 7=SDR/DDR/QDR

```
-bounce
```

Brings the active links down and back up if a new value is set.

Sample Output:

```
-> ismChassisSetSpeed 1
L01P01  link speed is SDR
L01P02  link speed is SDR
L01P03  link speed is SDR
...
```

> NOTE: This command can be used to set the supported link speed on each of the chassis ports. This command does not cause the associated ports to change the active or enabled speed of the port. The active and enabled speed is changed the next time the port transitions down and back up. Each external port must be connected to another similarly configured port to establish a link.
> Calling this function without a speed designator displays the current values of the port.
> Calling this function with the [-bounce] option disruptively brings active links down and back up so the links use the new setting immediately. Otherwise the new setting is used the next time links retrain.

ismPortEnable

Allows the user to enable or disable a port.

Syntax:

```
ismPortEnable port
```

Options:

```
port
```

Port name.

Sample Output:

```
-> ismPortEnable S3BL08b
```

NOTE: The option portName is case sensitive.

ismChassisSetEnable

Allows the user to view and enable ports for entire chassis.

Syntax:

```
ismChassisSetEnable [enable]
```

Options:

```
enable
```
1=enable, 0=disable.

Sample Output:

```
-> ismChassisSetEnable
L01P01  is ENABLED
L01P02  is ENABLED
L01P03  is ENABLED
L01P04  is ENABLED
L01P05  is ENABLED
L01P06  is ENABLED
L01P07  is ENABLED.
```

NOTE: Using this command without the enable option displays the current value of the port(s). When disabling ports, only cable ports are disabled.

ismPortDisable

Allows the user to enable or disable ports.

Syntax:

```
ismPortDisable port
```

Options:

```
port
```
Port name.

Sample Output:

```
-> ismPortDisable S3BL10a
```

NOTE: The port option is case sensitive.

ismPortSetSpeed

Allows the user to view and set port speeds.

Syntax:

```
ismPortSetSpeed port [speed] [-bounce]
```

Options:

```
port
```
Port name.
```
speed
```
1=SDR 2=DDR 3=SDR/DDR 4=QDR 5=SDR/QDR 6=DDR/QDR 7=SDR/DDR/QDR.
```
bounce
```
Brings the active links down and back up if a new value is set.

Sample Output:

```
-> ismPortSetSpeed S3BL08b 1
S3BL08b link speed is SDR.
```

NOTE: The port option is case sensitive. This command can be used to set
the supported link speed on the port. This command does not cause
the port to change the active or enabled speed of the port. The active
and enabled speed is changed the next time the port transitions down
then back up. Calling this function with only the portName option
displays the current values. Calling this function with the [-bounce]
option disruptively brings active links down then back up so the links
use the new setting immediately. Otherwise the new setting is used
the next time links retrain.

ismIslSet12x

Allows the user to view, set and unset the chassis link width (except external ports).

Syntax:

```
ismIslSet12x [enable8xBit] [-bounce]
```

Options:

```
enable8xBit
```
2=enable(8X only), 1=enable(AUTO 1X/4X/8X), 0=disable
```
-bounce
```
Brings the active links down and back up if a new value is set.

Sample Output:

```
-> ismIslSet12x 0
L01S2Ba 8x mode is DISABLED
L01S2Bb 8x mode is DISABLED
L01S2Bc 8x mode is DISABLED
...
```

NOTE: This command is included for backwards compatibility. The
ismIslSetWidth command is prefered.
This command ONLY works on chassis' which support 8X link
aggregation. This command will enable 8X link aggregation so that
INTERNAL switching will be at 8X. External ports will remain
configured to 4X and will not require special cabling configuration to
function correctly. To force links to only operate at 8X, use the '8X
only' option. Calling this function without an enable8xBit will display
the ports' current values.
Calling this function with the [-bounce] option disruptively brings
active links down and back up so the links use the new setting
immediately. Otherwise the new setting will be used the next time
links retrain.

ismChassisSetMtu

Allows the user to view, set and unset the chassis maximum packet MTU Capability and VL Capability
for all ports.

Syntax:

```
ismChassisSetMtu [mtuCap] {vlCap} [-bounce]
```

Options:

```
mtuCap
```
   4=2048 bytes, 5=4096 bytes.
```
vlCap
```
   1=VL0, 2=VL0,VL1, 3=VL0-VL3, 4=VL0-VL7
```
-bounce
```
   Brings the active links down and back up if a new value is set.

Sample Output:

```
-> 1-> ismChassisSetMtu
L01P01  MTUCap=4(2048 bytes) VLCap=4(8 VLs)
L01P02  MTUCap=4(2048 bytes) VLCap=4(8 VLs)
L01P03  MTUCap=4(2048 bytes) VLCap=4(8 VLs)
L01P04  MTUCap=4(2048 bytes) VLCap=4(8 VLs)
L01P05  MTUCap=4(2048 bytes) VLCap=4(8 VLs)
L01P06  MTUCap=4(2048 bytes) VLCap=4(8 VLs)
L01P07  MTUCap=4(2048 bytes) VLCap=4(8 VLs)
L01P08  MTUCap=4(2048 bytes) VLCap=4(8 VLs)
L01P09  MTUCap=4(2048 bytes) VLCap=4(8 VLs)
L01P10  MTUCap=4(2048 bytes) VLCap=4(8 VLs)
L01P11  MTUCap=4(2048 bytes) VLCap=4(8 VLs)
L01P12  MTUCap=4(2048 bytes) VLCap=4(8 VLs)
```

NOTE: This command will set the MTU capability for both internal and external switching. If the vlCap option is not specified, the command defaults to the maximumVL(s) for the selected mtuCap. Calling this function without an mtuCap designator displays the current value for each port.
Calling this function with the [-bounce] option disruptively brings active links down and back up so the links use the new setting immediately. Otherwise the new setting will be used the next time links retrain.

ismIslSetSpeed

Allows the user to view and set internal port link speeds.

Syntax:

```
ismIslSetSpeed [speed] [bounce]
```

Options:

speedFlag

   1=SDR 2=DDR 3=SDR/DDR 4=QDR 5=SDR/QDR 6=DDR/QDR 7=SDR/DDR/QDR

bounce

   Brings the active links down and back up if a new value is set.

Sample Output:

```
-> ismIslSetSpeed 2
L01S3Ba link speed is DDR
L01S1Ba link speed is DDR
L01S1Bb link speed is DDR
...
```

NOTE: This command sets the supported link speed on each of the internal switch-to-switch ports. This command does not cause the associated ports to change the active or enabled speed of the port. The active and enabled speed is changed the next time the port transitions down and back up. Calling this function without a parameter displays the current values of the port. Calling this function with the [-bounce] option disruptively brings active links down and back up so the links use the new setting immediately. Otherwise the new setting is used the next time links retrain.

ismShowPStatThresh

Displays the port statistic thresholds.

Syntax:

```
ismShowPStatThresh
```

Options:

```
None.
```

Sample Output:

```
-> ismShowPStatThresh
        Field            Threshold  Time Unit
        -----            ---------  ---------
 1         portXmitData:        0   (Percent of Max)
 2         portRecvData:        0   (Percent of Max)
 3         portXmitPkts:        0   (Percent of Max)
 4         portRecvPkts:        0   (Percent of Max)
 5         portXmitWait:        0   (1 Second)
 6        portSymbolErr:        0   (1 Second)
 7      portLinkErrRecv:        0   (1 Second)
 8       portLinkDowned:        0   (1 Second)
 9          portRecvErr:        0   (1 Second)
10    portRecvRemPhysErr:        0   (1 Second)
12      portXmitDiscard:        0   (1 Second)
13       portPKeyViolIn:        0   (1 Second)
14      portPKeyViolOut:        0   (1 Second)
15        portRawViolIn:        0   (1 Second)
16       portRawViolOut:        0   (1 Second)
17    portLocalLinkInteg:        0   (1 Second)
18  portExcBufferOverrun:        0   (1 Second)
19 portRelayedVL15Dropped:        0   (1 Second)
20   portLocalVL15Dropped:        0   (1 Second)
21     portNonSMPDropped:        0   (1 Second)
```

NOTE: The following are the available port statics thresholds:

■ 1 portXmitData: 32-bit data words transmitted

■ 2 portRecvData: 32-bit data words received

- 3 portXmitPkts: data packets transmitted

- 4 portRecvPkts: data packets received

- 6 portSymbolErr: a 8B10B encoding violation, or a displarity violation was detected

- 7 portLinkErrRecv: link error recovery process happened successfully

- 8 portLinkDowned: link error recovery process failed

- 9 portRecvErr: errors received

- 10 portRecvRemPhysErr: remote physical errors received

- 12 portXmitDiscard: port transmit discards

- 13 portPKeyViolIn: PKey inbound was invalid

- 14 portPKeyViolOut: PKey outbound was invalid

- 15 portRawViolIn: raw inbound packet discarded

- 16 portRawViolOut: raw outbound packet discarded

- 17 portLocalLinkInteg: link integrity errors

- 18 portExcBufferOverrun: excessive buffer overrun errors

- 19 portRelayedVL15Dropped: remote VL15 packet was dropped

- 20 portLocalVL15Dropped: local VL15 packet was dropped

- 21 portNonSMPDropped: non SMP packet was dropped

ismSetPStatThresh

Modifies the port statistic thresholds.

Syntax:

```
ismSetPStatThresh field threshold
```

Options:

```
-field
```
The name of the port status threshold field.

```
-threshold
```
The numeric threshold value.

Sample Output:

```
-> ismSetPStatThresh  portXmitData 0
successfully set port stat threshold portXmitData
```

NOTE: The available thresholds are:

- portXmitData
- portRecvData
- portXmitPkts
- portRecvPkts
- portSymbolErr
- portLinkErrRecv
- portLinkDowned
- portRecvErr
- portRecvRemPhysErr
- portXmitDiscard
- portPKeyViolIn
- portPKeyViolOut
- portRawViolIn
- portRawViolOut
- portLocalLinkInteg
- portExcBufferOverrun
- portRelayedVL15Dropped
- portLocalVL15Dropped
- portNonSMPDropped

ismRemoveStateDump

Removes switch ASIC state dumps files.

Syntax:

```
ismRemoveStateDump [all] [-leaf X] [-spine X]
```

Options:

```
all
```
Removes all switch ASIC state dump files.
```
-leaf X
```
Removes a switch ASIC state dump file for a specific leaf X.
```
-spine X
```
Removes the switch ASIC state dump files for a specific spine X.

Sample Output:

```
-> ismRemoveStateDump all
```

ismRemoveStateDump (12300)

Removes switch ASIC state dumps files.

Syntax:

```
ismRemoveStateDump
```

Options:

```
None
```

ismShowStateDump

Shows contents of switch ASIC state dumps.

Syntax:

```
ismShowStateDump [all] [-leaf X] [-spine X]
```

Options:

```
all
    Removes all switch ASIC state dump files.
-leaf X
    Removes a switch ASIC state dump file for a specific leaf X.
-spine X
Removes the switch ASIC state dump files for a specific spine X.
```

Sample Output:

```
-> ismRemoveStateDump all
```

ismShowStateDump (12300)

Shows contents of switch ASIC state dumps.

Syntax:

```
ismShowStateDump [all]
```

Options:

```
all
```
Removes all switch ASIC state dump files.

Sample Output:

```
-> ismRemoveStateDump all
```

ismPortSetWidth

View/Set/Unset the port link width for a specific port[s].

Syntax:

```
ismPortSetWidth [portName] [linkWidth] [-bounce]
```

Options:

```
portName
```
Valid port name. For ex. L101P01 or S201AP13L101P22

```
linkWidth
```
1=1X, 2=4X, 3=1X/4X, 4=8X, 5=1X/8X, 6=4X/8X, 7=1X/4X/8X

```
-bounce
```
Brings the active links down and back up if a new value is set.

Sample Output:

```
-> ismPortSetWidth L01P01 6
```

> NOTE: This command only works on platforms supporting 8X link aggregation. This command sets link aggregation to the specified width. The option **portName** is case sensitive. If this option contains spaces, it must be bounded by quotation marks (e.g., **ismPortSetWidth "Cable 1", 1**. Calling this function with only the **portName** option will display its current values.
> Calling this function with the [-bounce] option disruptively brings active links down and back up so the links use the new setting immediately. Otherwise the new setting will be used the next time links retrain.

ismChassisSetWidth

View/Set/Unset chassis link width (including external ports).

Syntax:

```
ismChassisSetWidth [linkWidth] [-bounce]
```

Options:

```
linkWidth
```

1=1X, 2=4X, 3=1X/4X, 4=8X, 5=1X/8X, 6=4X/8X, 7=1X/4X/8X

```
-bounce
```

Brings the active links down and back up if a new value is set.

Sample Output:

```
-> ismChassisSetWidth
L101P01 link width supported: 4X
L101P02 link width supported: 4X
L101P03 link width supported: 4X
L101P04 link width supported: 4X
L101P05 link width supported: 4X
L101P06 link width supported: 4X
L101P07 link width supported: 4X
L101P08 link width supported: 4X
L101P09 link width supported: 4X
L101P10 link width supported: 4X
L101P11 link width supported: 4X
L101P12 link width supported: 4X
L101P13 link width supported: 4X
L101P14 link width supported: 4X
L101P15 link width supported: 4X
L101P16 link width supported: 4X
....
```

NOTE: This command ONLY works on switches supporting 8X link
aggregation. This command sets link aggregation so that Internal
AND External switching will be at the specified width. Each 8X port
is 2 aggregated 4X ports (i.e. a DUO), where one will be reported as
a 8X port, and the other port is reported as disabled. Each of the two
aggregated 4X ports must be connected to another enabled 8X Duo
to establish an 8X link. To force links to only operate at 8X, use the
'8X' option. Calling this function without a parameter will display the
ports' current values.
Calling this function with the [-bounce] option disruptively brings
active links down and back up so the links use the new setting
immediately. Otherwise the new setting will be used the next time
links retrain.

ismIslSetWidth

View/Set/Unset chassis link width (except external ports).

Syntax:

```
ismIslSetWidth [linkWidth] [-bounce]
```

Options:

```
[linkWidth]:
```

1=1X, 2=4X, 3=1X/4X, 4=8X, 5=1X/8X, 6=4X/8X, 7=1X/4X/8X.

```
-bounce
```

Brings the active links down and back up if a new value is set.

Sample Output:

```
-> ismIslSetWidth 4
```

> NOTE: This command ONLY works on chassis' supporting 8X link aggregation. This command sets link aggregation so INTERNAL switching is the specified width. External ports remain configured to 4X and will not require special cabling configuration(s) to function correctly. Calling this function without any parameter displays the current values of the port(s).
> Calling this function with the [-bounce] option disruptively brings active links down and back up so the links use the new setting immediately. Otherwise the new setting will be used the next time links retrain.

ismPortSetBeacon

Allows the user to view and set the port beacon.

Syntax:

```
ismPortSetBeacon [port] [beacon]
```

Options:

```
port
```

Port name.

```
beacon
```

1=on, 0=off

Sample Output:

```
-> ismPortSetBeacon L101 on
```

> NOTE: The **port** optoin is case sensitive. Calling this function with only the **portName** option will only display the current values.

ismPortQsfpInfo

Displays information from installed Quad Small Form-factor pluggable (QSFP) cables

Syntax:

```
ismPortQsfpInfo [scope] [output option]
```

Options:

```
scope
```
   leafname, portname, or all
```
output option
```
   -verbose = display multi-line output

Sample Output:

```
-> ismPortQsfpInfo L101 -verbose
```

> NOTE: The Scope argument is case-sensitive and should adhere to the following formats:
>
> ■ all   = display info for all installed QSFPs in chassis
>
> ■ Lxxx   = display info for all installed QSFPs on leaf xxx
>
> ■ LxxxPyy = display info for QSFP installed on leaf xxx, port yy
>
> ■ Cableyy = display info for QSFP installed on port yy (12300 only)
>
> QSFP information may not be available for DDR hybrid cable assemblies. For 0.5 meter cables, the length may be reported as "0 meter".

ismChassisBounce

Bounce enabled ports for an entire chassis.

Syntax:

```
ismChassisBounce
```

Options:

```
None.
```

Sample Output:

```
-> ismChassisBounce
```

ismIslBounce

Bounce only the enabled internal ports.

Syntax:

```
ismIslBounce
```

Options:

```
None.
```

Sample Output:

```
-> ismIslBounce
```

ismPortBounce

Bounce enabled port(s).

Syntax:

```
ismPortBounce
```

Options:

```
None.
```

Sample Output:

```
-> ismPortBounce
```

TimeManagement

time

Configure the time on the device.

Syntax:

```
time -S ipaddr | -T hhmmss[mmddyyyy]
```

Options:

```
-S ipaddr
```

Sets the NTP Server IP address.

```
-T hhmmss[mmddyyyy]
```

Set the local clock time hour, minutes, and seconds along with the month, day, and year.

Sample Output:

```
-> time -S 172.26.0.254
Configured the NTP server ip address successfully
13:53:02  10/06/2006
Configured to use NTP server IP address: 172.26.0.254
```

> NOTE: Time can be configured locally (using a local clock) or set to be updated by an SNTP server. Please note that if you set the time locally, the unit will unconfigure the NTP server IP address if set. If no options are passed in, the current system time will be printed out.

timeZoneConf

Display/configure the time zone setting.

Syntax:

```
timeZoneConf [offset]
```

Options:

```
offset
```

The time offset in relation to Greenwich Mean Time (GMT).

Sample Output:

```
-> timeZoneConf -5
Timezone offset successfully configured
Current time zone offset is: -5
```

NOTE: The offset parameter specifies a time zone the system should use when setting the time. In the U.S. the following time zones are in effect: Eastern Standard Time = GMT -5; Central Standard Time = GMT -6; Mountain Standard Time = GMT -7; Pacific Standard Time = GMT -8. GMT = GreenwichMean Time..

timeDSTConf

Configure and display the Daylight Saving Time settings.

Syntax:

```
timeDSTConf [sw sd sm ew ed em]
```

Options:

sw

Start which, valid values: 1 = 1st, 2 = 2nd, 3 = 3rd, 4 = 4th, 5 = 5th

sd

Start day, valid values: 1 = Sunday, 2 = Monday, 3 = Tuesday, 4 = Wednesday, 5 = Thursday, 6 = Friday, 7 = Saturday

sm

Start month, valid values: 3 = March, 4 = April, 5 = May, 6 = June, 7 = July, 8 = August, 9 = September, 10 = October, 11 = November

ew

End which, valid values: 1 = 1st, 2 = 2nd, 3 = 3rd, 4 = 4th, 5 = 5th

ed

End day, valid values: 1 = Sunday, 2 = Monday, 3 = Tuesday, 4 = Wednesday, 5 = Thursday, 6 = Friday, 7 = Saturday

em

End month, valid values: 3 = March, 4 = April, 5 = May, 6 = June, 7 = July, 8 = August, 9 = September, 10 = October, 11 = November

Sample Output:

```
-> timeDSTConf 2 1 3 1 1 11
Timezone offset successfully configured
Current DST = Start: 2'nd Sunday of March End: 1'st Sunday of November
```

NOTE: Example: To set the daylight savings time to start on the 1st Sunday of April and end on the 4th Sunday of October the command would be: 'timeDSTConf 1 1 4 4 1 10'.

timeNtpTimeout

Display or set the number of seconds to wait for a NTP response.

Syntax:

```
timeNtpTimeout numSeconds
```

Options:

```
numSeconds
```
New timeout setting.

Sample Output:

```
-> timeNtpTimeout
Current NTP timeout value: 2 seconds
-> timeNtpTimeout 3
Current NTP timeout changed to 3 seconds
```

NOTE: With no arguments, this command displays the current NTP timeout settings. This is the amount of time (in seconds) for the system to wait for a response from the NTP server. This setting can be configured by using the same command with the new timeout value (in whole seconds) as the only argument. The default settings is 2 seconds. The NTP timeout value is not used on line cards or slave CMUs.

timeNtpRefreshTime

Display or set the delay between syncing the clock via NTP.

Syntax:

```
timeNtpRefreshTime numSeconds
```

Options:

```
numSeconds
```
New refresh delay setting.

Sample Output:

```
-> timeNtpRefreshTime
Current NTP refresh delay value: 60 seconds
-> timeNtpRefreshTime 50
Current NTP refresh delay changed to 50 seconds.
```

NOTE: With no arguments this command displays the current NTP refresh time, which is the delay in seconds between attempts to sync the clock via NTP. This value can be configured by using this same command with the new refresh time (in whole seconds) as the only argument. The NTP refresh time is not used on line cards or slave CMUs.

SNMP

snmpCommunityConf

Displays and allows the changing of the SNMP community strings.

Syntax:

```
snmpCommunityConf [-r readonly_comm_str]
[-w read_write_comm_str]
```

Options:

```
-r
```
A read-only community string.

```
-w
```
A read/write community string.

Sample Output:

```
-> snmpCommunityConf -r public
Read Only Community String Was Set To: public
```

> NOTE: If no options are entered the current SNMP read-only community string and read-write community string is displayed. If options are entered then the corresponding community string is changed.

snmpTargetAddr

Displays and allows the changing of the **snmpTargetAddrTable** entries.

Syntax:

```
snmpTargetAddr [show | delete | add | edit] -n name
[-a addr] [-p port] [-t timeout] [-r retry_count]
[-l tag_list] [-v parameters] [-s storage_type] [-i
status]
```

Options:

show

Displays the contents of the **snmpTargetAddrTable**.

add

Adds a row to the **snmpTargetAddrTable**.

edit

Modifies an existing row in the **snmpTargetAddrTable**.

delete

Removes an existing row of the **snmpTargetAddrTable**.

-n

Name. A unique name used to identify a row. Any name with a space (e.g. xxx v3) must be surrounded by double quotes (").

-a

Address. The target machine IP address in dotted decimal form.

-p

Port. The target port to send traps and information.

-t

Timeout. The time to wait for an information response.

-r

Retry count. The number of re-send attempts for information.

-l

Tag list. Indicates the traps and information that is sent.

-v

Parameters. This maps to an entry in the **snmpTargetAddrTable.**

-s

Storage type. Determines whether the entry is saved in flash memory.

-i

Status, 1=Active, 2=Not In Service, 3=Not Ready.

Sample Output:

```
-> snmpTargetAddr
rfc2573t:snmpTargetAddrTDomain: nms v1 : 1.3.6.1.6.1.1
rfc2573t:snmpTargetAddrTDomain: nms v2 : 1.3.6.1.6.1.1
rfc2573t:snmpTargetAddrTDomain: nms v3 : 1.3.6.1.6.1.1
rfc2573t:snmpTargetAddrTAddress: nms v1 :  (ip addr)00.00.00.00 (port)0000
rfc2573t:snmpTargetAddrTAddress: nms v2 :  (ip addr)00.00.00.00 (port)0000
rfc2573t:snmpTargetAddrTAddress: nms v3 :  (ip addr)00.00.00.00 (port)0000
```

NOTE: The output is in the form: 'mib : mib_object : table_index : value' . For more details on the snmpTargetAddrTable see SNMP-TARGET-MIB, RFC 2573.

snmpTargetParams

Displays the **snmpTargetParamsTable** entries.

Syntax:

```
snmpTargetParams [show]
```

Options:

```
show
```

Displays the contents of the **snmpTargetParamsTable**.

Sample Output:

```
-> snmpTargetParams
rfc2573t:snmpTargetParamsMPModel: v1 params : 0
rfc2573t:snmpTargetParamsMPModel: v2 params : 1
rfc2573t:snmpTargetParamsMPModel: v3 params : 3
rfc2573t:snmpTargetParamsSecurityModel: v1 params : 1
rfc2573t:snmpTargetParamsSecurityModel: v2 params : 2
rfc2573t:snmpTargetParamsSecurityModel: v3 params : 3
```

> NOTE: The output is in the form: 'mib : mib_object : table_index : value' . For more details on the snmpTargetParamsTable see SNMP-TARGET-MIB, RFC-2573.

snmpNotifyProfile

Displays the **snmpNotifyFilterProfileTable** entries.

Syntax:

```
snmpNotifyProfile [show]
```

Options:

```
show
```

Displays the contents of the **snmpNotifyFilterProfileTable**.

Sample Output:

```
-> snmpNotifyProfile
rfc2573n:snmpNotifyFilterProfileName: v1 params : v1 params
rfc2573n:snmpNotifyFilterProfileName: v2 params : v2 params
rfc2573n:snmpNotifyFilterProfileName: v3 params : v3 params
rfc2573n:snmpNotifyFilterProfileStorType: v1 params : 3
rfc2573n:snmpNotifyFilterProfileStorType: v2 params : 3
rfc2573n:snmpNotifyFilterProfileStorType: v3 params : 3
```

NOTE: The output is in the form: 'mib : mib_object : table_index : value' . For more details on the snmpNotifyFilterProfileTable see SNMP-NOTIFICATION-MIB, RFC-2573.

snmpNotifyFilter

Displays the **snmpNotifyFilterTable** entries.

Syntax:

```
snmpNotifyFilter [show]
```

Options:

```
show
```
Displays the contents of the **snmpNotifyFilterTable**.

Sample Output:

```
-> snmpNotifyFilter
rfc2573n:snmpNotifyFilterMask:  v1 params : 0
rfc2573n:snmpNotifyFilterMask:  v2 params : 0
rfc2573n:snmpNotifyFilterMask:  v3 params : 0
rfc2573n:snmpNotifyFilterType:  v1 params : 1
rfc2573n:snmpNotifyFilterType:  v2 params : 1
rfc2573n:snmpNotifyFilterType:  v3 params : 1
```

NOTE: The output is in the form: 'mib : mib_object : table_index : value' . For more details on the snmpNotifyFilterTable see SNMP-NOTIFICATION-MIB, RFC-2573.

snmpNotify

Displays the **snmpNotifyTable** entries.

Syntax:

```
snmpNotify [show]
```

Options:

```
show
```
Displays the contents of the **snmpNotifyTable**.

Sample Output:

```
-> snmpNotify
rfc2573n:snmpNotifyTag: switch : rfc1493
rfc2573n:snmpNotifyTag: interfaces : rfc2233
rfc2573n:snmpNotifyTag: rmon : rfc1757
rfc2573n:snmpNotifyTag: snmp : rfc1907
rfc2573n:snmpNotifyTag: tms : tmscom
rfc2573n:snmpNotifyType: switch : 1
```

NOTE: The output is in the form: 'mib : mib_object : table_index : value' . For
more details on the snmpNotifyTable see
SNMP-NOTIFICATION-MIB, RFC-2573.

snmpSystem

Displays and allows the changing of the Mib-2 system entries.

Syntax:

```
snmpSystem [show | edit] [-n sysName] [-l sysLocation]
[-c sysContact]
```

Options:

show

show the contents of the snmpTargetAddrTable.

edit

modify an existing row in the snmpTargetAddrTable.

-n

sysName.

-c

sysContact.

-l

sysLocation.

Sample Output:

```
-> snmpSystem show
rfc1907:sysDescr: þ : 20.28.4D.61.73.74.65.72.29
rfc1907:sysObjectID:  : 1.3.6.1.4.1.10222.7.1.2
rfc1907:sysUpTime:  : 1 Day(s), 23 Hour(s), 34 Minute(s), 47 Second(s)
rfc1907:sysContact: þ : {no value}
rfc1907:sysName: þ : QLogic 12800-040-254
rfc1907:sysLocation: þ : Main Chassis Unit, Slot 254
rfc1907:sysServices:  : 79
```

NOTE: The output is in the form: 'mib : mib_object : table_index : value'.

snmpUsrSec

Configure and display SNMP V3 users.

Syntax:

```
snmpUsrSec [add | show | edit | delete] [name] [-a algo
[key]]
```

Options:

```
add
```
add an entry to the V3 user table.
```
show
```
show entries in the V3 user table.
```
edit
```
modify an entry in the V3 user table.
```
delete
```
remove an entry in the V3 user table.
```
-a
```
authentication algorithm and key. Options: MD5, SHA, NONE. Passphrase is required unless NONE is specified.

Sample Output:

```
-> snmpUsrSec
User    : initialmd5
Auth    : MD5
Auth Key: 0x047b473f93211a17813ce5fff290066b
Priv    : NONE

User    : initialsha
Auth    : SHA
Auth Key: 0x1c8cbd687fb0f0a22ddd24315db0d84c09eb5ff3
Priv    : NONE

User    : initialnone
Auth    : NONE
Priv    : NONE
```

NOTE: Handles configuration and display of SNMP v3 users. Supported authentication algorithms are: NONE, MD5, and SHA. A passphrase is required for all except the NONE algorithm. No privacy algorithms are currently supported.

CaptureInfo

capture

Displays information for this switch.

Syntax:

```
capture
```

Options:

```
None.
```

> NOTE: This command is intended for support personnel.

captureFw

Display firmware information for this device.

Syntax:

```
captureFw
```

Options:

```
None.
```

> NOTE: This command is intended for support personnel.

captureLog

Display log information for this device.

Syntax:

```
captureLog
```

Options:

```
None.
```

> NOTE: This command is intended for support personnel.

captureSm

Displays Subnet Management information for this device.

Syntax:

```
captureSm
```

Options:

```
None.
```

NOTE: This command is intended for support personnel.

captureIsm

Display Infiniband switch information for this device.

Syntax:

```
captureIsm
```

Options:

```
None.
```

NOTE: This command is intended for support personnel.

captureChassis

Display chassis information for this device.

Syntax:

```
captureChassis
```

Options:

```
None.
```

NOTE: This command is intended for support personnel.

captureNetwork

Display chassis information for this device.

Syntax:

```
captureNetwork
```

Options:

> None.

> NOTE: This command is intended for support personnel.

captureMisc

Display miscellaneous information for this device.

Syntax:

```
captureMisc
```

Options:

> None.

> NOTE: This command is intended for support personnel.

captureSnmp

Display SNMP information for this device.

Syntax:

```
captureSnmp
```

Options:

> None.

> NOTE: This command is intended for support personnel.

captureShell

Display shell command information for this device.

Syntax:

```
captureShell
```

Options:

> None.

> NOTE: This command is intended for support personnel.

QLOGIC
The Ultimate in Performance

Corporate Headquarters QLogic Corporation 26650 Aliso Viejo Parkway   Aliso Viejo, CA 92656 949.389.6000   **www.qlogic.com**

International Offices UK | Ireland | Germany | India | Japan | China | Hong Kong | Singapore | Taiwan